

Digital Surfaces of Revolution made simple

Éric Andrès, Gaëlle Largeteau-Skapin

► **To cite this version:**

Éric Andrès, Gaëlle Largeteau-Skapin. Digital Surfaces of Revolution made simple. DGCI 2016: 19th IAPR international conference on Discrete Geometry for Computer Imagery, Apr 2016, Nantes, France. Springer Verlag, 9647, Discrete Geometry for Computer Imagery 19th IAPR International Conference, DGCI 2016, Nantes, France, April 18-20, 2016. Proceedings. <hal-01289781>

HAL Id: hal-01289781

<https://hal.archives-ouvertes.fr/hal-01289781>

Submitted on 17 Mar 2016

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.

DRAFT : Digital Surfaces of Revolution made simple

Eric Andres and Gaelle Largeteau-Skapin

Université de Poitiers, Laboratoire XLIM, SIC, UMR CNRS 7252, BP 30179, F-86962
Futuroscope Chasseneuil, France

{eric.andres,gaelle.largeteau.skapin}@univ-poitiers.fr

Abstract. In this paper we present a new, simple, method for creating digital 3D surfaces of revolution. One can choose the topology of the surface that may have 0, 1 or no tunnels. The definition we propose is not limited to circles as curves of revolution but can be extended to any type of implicit curves.

Keywords: digital surfaces, implicit functions, surface of revolution

1 Introduction

Pottery wheels made the surface of revolution one of the first types of complex surfaces that man has created. In Computer Graphics [1], surfaces of revolution, as special case of swept surfaces, have always represented a simple way of constructing surfaces that appear quite frequently in real life: glasses, chess pieces, lamps, flower pots, etc. They are based on a 2D curve profile called *generatrix* and a *revolution curve* which is classically a 2D circle. In digital geometry, there are not many papers that have dealt specifically with the problem of generating digital surfaces of revolution. There is the works of Nilo Stolte that indirectly looked at such problems as part of research on the visualization of implicit surfaces with cylindrical coordinates [2, 3]. In two more recent papers, G.Kumar and P.Bhowmick [4, 5] proposed a virtual pottery design tool based on digital surfaces of revolution. Their idea is to generate a digital surface of revolution by superposing 2D digital annuli. The horizontal section of a continuous surface of revolution is, by definition, the curve of revolution and thus, in general, a circle. A vertical slice of height one can of course be very complicated but rasterized, it corresponds to a 2D digital annulus. The authors work with a classical notion of circle close to the one proposed by Bresenham, defined only for integer radii and center, and this creates some difficulties. While the center of the annulus for z is the point of integer coordinates $(0, 0, z)$, there is no particular reason for the interior or exterior radii to be an integer. Moreover, Bresenham circles do not fill space (i.e. concentric Bresenham circles of increasing radii leave points that do not belong to any circle). G.Kumar and P.Bhowmick solve this by determining the missing (also sometimes called *absentee*) voxels that would otherwise leave 6-connected holes in the revolution surface. Their method is quite general since

it does not require the generatrix to be functional in z (only one ordinate value per z , if we consider the generatrix to be in the yz -space). The method for filling the holes with absentee voxels is however quite complicated and limited to circular curves of revolution. One way around this problem could be to use Andres circles [6–8] that, do, fill space and are defined for arbitrary center and radii.

In this paper we are going to adopt a different approach based on the digitization of implicit curves and surfaces proposed at the last DGCI conference [9] with the participation of the authors of the present paper. We showed that it is possible to analytically define digital implicit surfaces in dimension n with controlled topology. Under some regularity conditions [9], one can ensure that the analytically defined digital implicit surface is a k -tunnel free $(n-1)$ -dimensional digital surface in dimension n , with $0 \leq k < n$. In this paper, we show how this can be used to create surfaces of revolution with a 2D explicit function as generatrix and a 2D implicit curve as curve of revolution. The generatrix and the curve of revolution are combined in order to obtain a 3D implicit surface that is then digitized into 0, 1 or 2-tunnel free digital surfaces of revolution. The control of the topology allows us to choose the type of surface we need for a given application. For instance, for visualization purposes, a thinner 2-tunnel free surface might suffice. If we want to use a 3D printer however, we need 2-connectivity (voxel-face connectivity) and thus 0 or 1-tunnel free surfaces of revolution. The curve of revolution is, in our model, not limited to a circle. One can use any implicit 2D curve as long as it separates space into a positive and a negative side. Of course, if the curve is infinite, one has to consider a generation window in which to consider the revolution curve or simply be satisfied with an analytical description of the surface of revolution. By considering only an explicit function generatrix of type $y = f(z)$, our generatrix are less general than [4, 5]. There are however solutions and we will give an example in the conclusion. There remains some work to be done in this regard. The method presented in this paper is very simple and general but it comes with some limitations when the curve of revolution does not respect some regularity conditions [9]. We will give an example of a surface of revolution that has holes.

In section two, we present basic notations and the analytical implicit surface digitization method. In section three, we present our digital surface of revolution, show some results and extensions. We finish the section with the limitations of our generation method. In the fourth and last section we conclude and present short and long term perspectives for this work.

2 Basic notions and notations

Let $\{e_1, \dots, e_n\}$ denote the canonical basis of the n -dimensional Euclidean vector space. Let \mathbb{Z}^n be the subset of \mathbb{R}^n that consists of all the integer coordinate points. A *digital (resp. Euclidean) point* is an element of \mathbb{Z}^n (resp. \mathbb{R}^n). We denote by x_i the i -th coordinate of a point or a vector x , that is its coordinate associated to e_i . A *digital (resp. Euclidean) object* is a set of digital (resp. Euclidean) points.

For all $k \in \{0, \dots, n-1\}$, two integer points v and w are said to be k -adjacent or k -neighbors, if for all $i \in \{1, \dots, n\}$, $|v_i - w_i| \leq 1$ and $\sum_{j=1}^n |v_j - w_j| \leq n - k$. In the 2-dimensional plane, the 0- and 1-neighborhood notations correspond respectively to the classical 8- and 4-neighborhood notations. In the 3-dimensional space, the 0-, 1- and 2-neighborhood notations correspond respectively to the classical 26-, 18- and 6-neighborhood notations.

A k -path is a sequence of integer points such that every two consecutive points in the sequence are k -adjacent. A digital object E is k -connected if there exists a k -path in E between any two points of E . A maximum k -connected subset of E is called a k -connected component. Let us suppose that the complement of a digital object E , $\mathbb{Z}^n \setminus E$ admits exactly two k -connected components F_1 and F_2 , or in other words that there exists no k -path joining integer points of F_1 and F_2 , then E is said to be k -separating, or k -tunnel free, in \mathbb{Z}^n . If there is no path from F_1 to F_2 then E is said to be 0-separating or simply separating.

Let \oplus be the Minkowski addition, known as dilation, such that $\mathcal{A} \oplus \mathcal{B} = \cup_{b \in \mathcal{B}} \{a + b : a \in \mathcal{A}\}$.

2.1 Implicit Surface Digitization

In this paper we are considering analytical digital implicit surfaces as defined in [9]. Let \mathcal{S} be an implicit surface $\mathcal{S} = \{x \in \mathbb{R}^n : f(x) = 0\}$ which separates space into one (or several) region(s) where $f(x) < 0$ and one (or several) region(s) where $f(x) > 0$. Our digitization method is based on a morphological type digitization method for implicit surfaces based on k -flakes (introduced in [10]). Those adjacency flakes can be described as the union of a finite number of straight segments centered on the origin (see figure 1).

Definition 1. Let $0 \leq k < 3$. The minimal k -adjacency flake, $F_k(\rho)$ with radius $\rho \in \mathbb{R}^+$ is defined by:

$$F_k(\rho) = \left\{ \lambda u : \lambda \in [0, \rho], u \in \{-1, 0, 1\}^3, \sum_{i=1}^3 |u_i| = 3 - k \right\}.$$

The Flake-digitization of a $(n-1)$ -dimensional surface S in \mathbb{R}^n is defined by $\mathcal{F}_k(S) = \{v \in \mathbb{Z}^n : (F_k(\frac{1}{2}) \oplus v) \cap S \neq \emptyset\}$. Under some conditions [9], the flake digitization can be analytically characterized by considering only the vertices of the k -flake. We are now simply going to consider that this analytical characterization corresponds to a proper analytical digitization method defined by:

$$\mathcal{A}_k(S) = \left\{ v \in \mathbb{Z}^n : \begin{array}{l} \min\{f(x) : x \in (v \oplus F_k(1/2))\} \leq 0 \\ \text{and } \max\{f(x) : x \in (v \oplus F_k(1/2))\} \geq 0 \end{array} \right\}.$$

When the regularity conditions are verified then $\mathcal{A}_k(S) = \mathcal{F}_k(S)$ otherwise there are some differences that may in some cases create topological problems (see section 3.6) but this is largely compensated by the fact that $\mathcal{A}_k(S)$ is easy to construct while $\mathcal{F}_k(S)$ may not. The regularity conditions as presented in [9] are

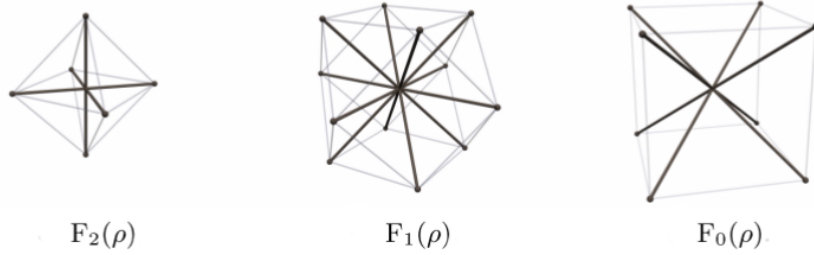


Fig. 1. Adjacency flakes in 3D.

the following: S should be r -regular [11] nD surface with $r > (\sqrt{n-k} + \sqrt{n})/2$. An r -regular set [11] is closed set such that for all $x \in \delta E$, it is possible to find two osculating open balls of radius r , one lying entirely inside E and the other lying entirely outside E . These conditions ensures the k -tunnel freeness of the digital implicit surface. They also preserve the connected components between the complement of the implicit surface and the complement of its digitization with regard to the k -adjacency relationship. These conditions are sufficient but not necessary. A necessary condition is still an open (and difficult) question. For short, $\mathcal{A}_k(S)$ is different from $\mathcal{F}_k(S)$ in places where the details of the curve S are small compared to the grid size which is not completely surprising. What is important is that the flake digitization $\mathcal{F}_k(S)$ is k -separating in \mathbb{Z}^n if it, basically, defines a positive and a negative region (for f defining S) greater than a pixel [10, 9]. This remains true for $\mathcal{A}_k(S)$ under the restrictions we mentioned.

3 Digital Surface of Revolution

3.1 Implicit surface of revolution definition

The surfaces of revolution are usually defined by rotating a curve (called the generatrix) around a straight line (the axis). In this paper, we generalise the notion: we use an explicit function $y = g(z)$ as a generatrix and instead of rotating it around an axis, we are using a *revolution* curve. In our work, the revolution function can be any $2D$ implicit continuous curve $r(x, y) = 0$, $(x, y) \in [-1, 1]^2 \subset \mathbb{R}^2$, that separates the window $[-1, 1]^2$ into two regions (one where the function r is positive, one where it is negative). If the curve of revolution is a circle then we obtain the classical surface of revolution. Note that the y of the generatrix definition $y = g(z)$ is not the same y that appears in the curve of revolution implicit equation $r(x, y) = 0$.

The surfaces we define use the generatrix as an homothetic factor for the revolution curve:

Definition 2.

$$S(g, r) = \left\{ (x, y, z) \in \mathbb{R}^3, r \left(\frac{x}{g(z)}, \frac{y}{g(z)} \right) = 0 \right\}$$

There is therefore a constraint on the generatrix: $\forall z \in \mathbb{R}, g(z) \neq 0$. Actually, if one considers surfaces of revolution, then one can restrict the definition to $g(z) \geq 1$. For $g(z) = 1$, with $r(x, y)$ defined on $[-1, 1]^2$, the slice of the digital surface of revolution corresponding to z will only be, at most, one voxel big. Smaller values of $g(z)$ may only make the slice disappear because it is smaller than what the analytical flake digitization can detect. Also, this constraint can be lifted in many cases by simply developing the expression r depending on how it is defined. For instance, when $r(x, y) = x^2 + y^2 - 1 = 0$ is the unit circle c , the resulting surface is the classical surfaces of revolution: $S(g, c) = \{x, y, z \in \mathbb{R}^3, x^2 + y^2 - g(z)^2 = 0\}$.

In all cases, we obtain an implicit surface that can be digitized using the \mathcal{A}_k -digitization.

3.2 Digital Generatrix

Sometimes, the generatrix we would like to consider is not defined by an explicit function but by a digital curve given as a set of digital points (such as an hand drawn generatrix for instance). In this case, one can easily recreate an explicit function generatrix.

There are several simple ways to transform a set of discrete points into an explicit function. The point to look out for is that the generatrix must be defined in such a way that it allows to compute values for $z \pm \frac{1}{2}$ (see section 3.3). A first method consist in creating an explicit function by interpolation of the discrete points that form your digital curve. A second method consists in simply consider that each integer z value is valid for the interval $[z - 1/2, z + 1/2]$.

The last method consists in decomposing the digital curve into digital line segments and then taking the continuous analog of the digital line: for a digital line $0 \leq ax - by + c < b$, we will consider the continuous line $ax - by - c - b/2 = 0$ or $ax - by - c - a/2 = 0$ depending on the orientation. A little care has to be taken for the end points of the different digital lines. If the intersections of two consecutive line segments does not fall in the pixel of the digital point corresponding to the common end point then a little patch function has to be added (see [12]) or one has to use an adapted line recognition where this problem does not occur [13]. This creates a piecewise defined explicit function. Figure 2 shows a digitized sinusoid that we have decomposed into digital line segments. In this example, the curve of revolution is a circle.

3.3 Algorithm

The digitization algorithm has been implemented in Mathematica. It is decomposed into two main functions: the first function tests if a point is in the digitization; the second scans the 3D digital space, applies the previous test and adds

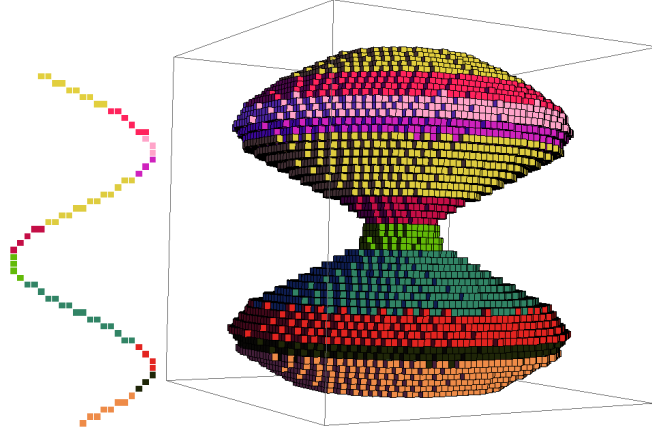


Fig. 2. A digital generatrix decomposed into digital straight segments (left) and the resulting revolution surface (right) using the unit implicit circle as revolution curve.

the valid points to the digitization result. We present here the three functions (one for each 3D flake digitization) that tests if a voxel $v(x, y, z)$ belongs to the digitization of the surface S defined by its implicit equation. Then we present the scanning function. Since the code for those functions is very simple, we present the functions as they appear in the mathematica notebook.

\mathcal{A}_2 -digitization test: This will define a surface that is 2-separating and 1-connected. This is what is classically called a *naive* surface in the digital geometry community. S is the implicit 3D surface. The result is a Boolean expression.

```

inNaif3D[S_, x_, y_, z_] :=
Module[{f1, f2, f3, f4, f5, f6},
  f1 = 1.0 * S[x + 0.5, y, z]; f2 = 1.0 * S[x - 0.5, y, z];
  f3 = 1.0 * S[x, y - 0.5, z]; f4 = 1.0 * S[x, y + 0.5, z];
  f5 = 1.0 * S[x, y, z + 0.5]; f6 = 1.0 * S[x, y, z - 0.5];
  And[Min[f1, f2, f3, f4, f5, f6] <= 0, Max[f1, f2, f3, f4, f5, f6] >= 0]];

```

\mathcal{A}_1 -digitization test: This will define a surface that is 1-separating and 0-connected. This is an intermediary surface type between the naive and the supercover types.

```

inInter3D[S_, x_, y_, z_] :=
Module[{f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12},

```

```

f1 = 1.0 * S[x + 0.5, y + 0.5, z]; f2 = 1.0 * S[x - 0.5, y + 0.5, z];
f3 = 1.0 * S[x + 0.5, y - 0.5, z]; f4 = 1.0 * S[x - 0.5, y - 0.5, z];
f5 = 1.0 * S[x + 0.5, y, z + 0.5]; f6 = 1.0 * S[x + 0.5, y, z - 0.5];
f7 = 1.0 * S[x - 0.5, y, z + 0.5]; f8 = 1.0 * S[x - 0.5, y, z - 0.5];
f9 = 1.0 * S[x, y + 0.5, z + 0.5]; f10 = 1.0 * S[x, y + 0.5, z - 0.5];
f11 = 1.0 * S[x, y - 0.5, z + 0.5]; f12 = 1.0 * S[x, y - 0.5, z - 0.5];
And[Min[f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12] <= 0,
Max[f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12] >= 0]];

```

\mathcal{A}_0 -digitization test: This will define a surface that is 0-separating and 0-connected. This is close to the classical supercover type surfaces [14–17].

```

inSuper3D[S_, x_, y_, z_] :=
Module[{f1, f2, f3, f4, f5, f6, f7, f8},
f1 = 1.0 * S[x + 0.5, y + 0.5, z + 0.5]; f2 = 1.0 * S[x + 0.5, y + 0.5, z - 0.5];
f3 = 1.0 * S[x + 0.5, y - 0.5, z + 0.5]; f4 = 1.0 * S[x + 0.5, y - 0.5, z - 0.5];
f5 = 1.0 * S[x - 0.5, y + 0.5, z + 0.5]; f6 = 1.0 * S[x - 0.5, y + 0.5, z - 0.5];
f7 = 1.0 * S[x - 0.5, y - 0.5, z + 0.5]; f8 = 1.0 * S[x - 0.5, y - 0.5, z - 0.5];
And[Min[f1, f2, f3, f4, f5, f6, f7, f8] <= 0,
Max[f1, f2, f3, f4, f5, f6, f7, f8] >= 0]];

```

Scan Algorithm: In the following module, (x_0, x_1) , (y_0, y_1) and (z_0, z_1) are the intervals on which to generate the surface. A list of elements in mathematica, for instance a, b, c , is defined by $\{a, b, c\}$. The operation 'AppendTo' adds a value at the end of the list. The result of this module will be a list of points (a point is defined as a list of three coordinates). 'TopoTest' stands for one of the above mentioned modules inNaif3D, inInter3D, inSuper3D.

```

Revolution [S_, x0_, x1_, y0_, y1_, z0_, z1_] :=
Module[{x, y, z, listpoint}, listpoint = {}];
For[x = x0, x <= x1, x ++,
For[y = y0, y <= y1, y ++,
For[z = z0, z <= z1, z ++,
If[TopoTest[S, x, y, z],
AppendTo[listpoint, {x, y, z}]];
]];
listpoint];

```

In order to be concise, the scan algorithm we present here is a very basic one. It is not very difficult to design one that works by propagation with a seed point and this for all three types of surfaces. A seed point is also not difficult to determine when considering surfaces of revolution.

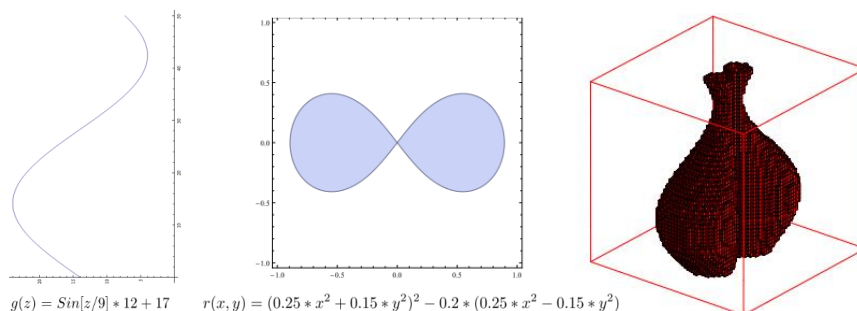


Fig. 3. Digital Generalized Surface of Revolution. The generatrix (left), the revolution curve (center) and the resulting digital surface (right).

3.4 Results

In this part, we present some examples of digital surfaces of revolution. Figure 3 presents the digitization of a generalized surface of revolution with a sinusoid function $g(z) = \text{Sin}[z/9] * 12 + 17$ as generatrix and the lemniscate function $r(x, y) = (0.25 * x^2 + 0.15 * y^2)^2 - 0.2 * (0.25 * x^2 - 0.15 * y^2)$ as revolution curve.

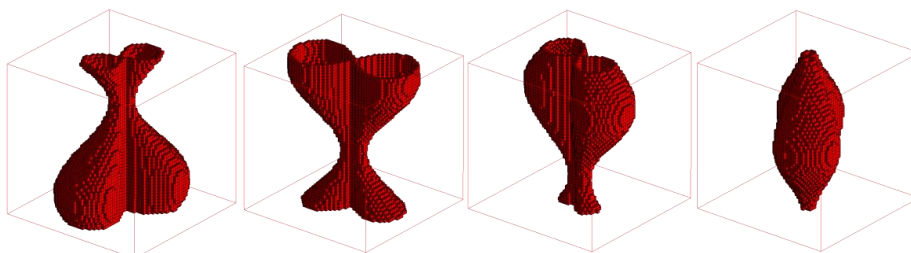


Fig. 4. Several examples of digital surfaces obtained with lemniscate revolution curves and sinusoidal generatrices.

Figure 4 presents similar objects than in figure 3. Between two objects, the lemniscate (the revolution curve) has been rotated and the phase of the sinusoid (the generatrix) has been changed¹. The view point of the camera is the same for the four objects.

Figure 5 presents the three 3D digitizations of a surface: the naive \mathcal{A}_2 -digitization on the top right, the \mathcal{A}_1 -digitization on the bottom left and the \mathcal{A}_0 -digitization on the bottom right. This surface is built using the same generatrix as in figure 3 and $r(x, y) = (2x^2 + 2y^2)^3 - 48x^2y^2 = 0$ as a revolution

¹ An animated version of these results can be seen on the following web page: [http : //xlim - sic.labo.univ - poitiers.fr/themes/ig/ig_axe_3.php](http://xlim-sic.labo.univ-poitiers.fr/themes/ig/ig_axe_3.php)

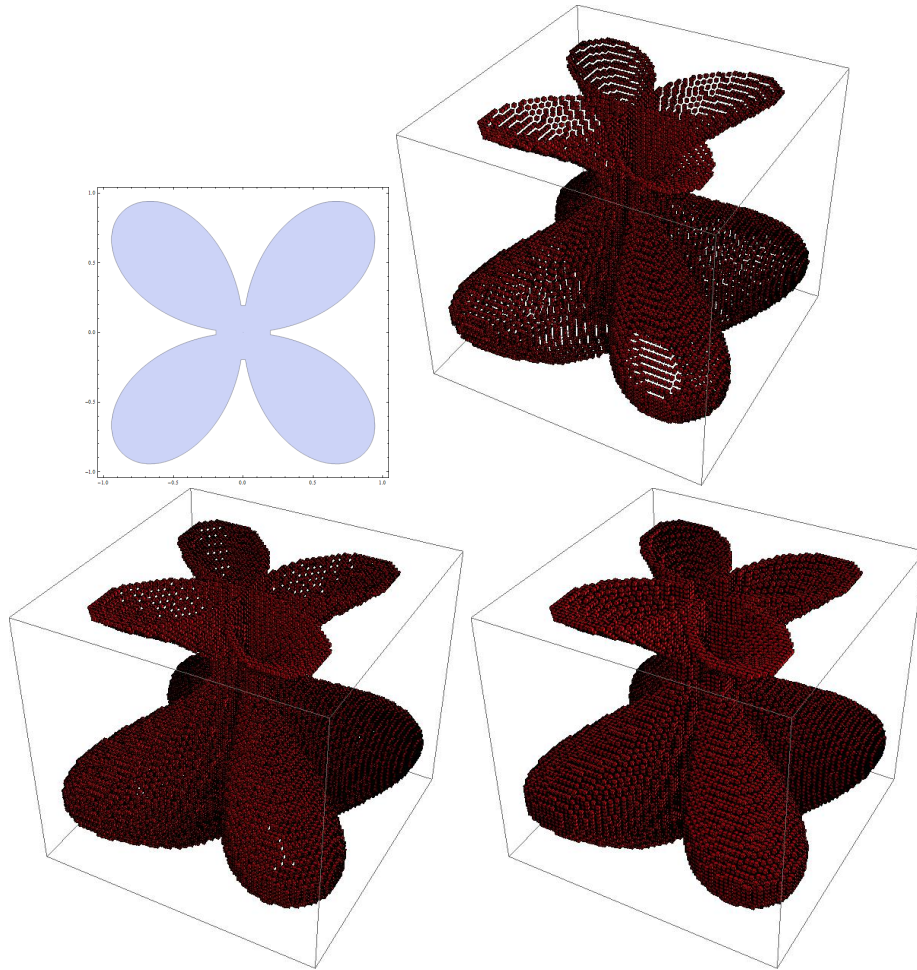


Fig. 5. The three different digitizations ($\mathcal{A}_2, \mathcal{A}_1$ and \mathcal{A}_0) with $r(x, y) : (2x^2 + 2y^2)^3 - 48x^2y^2 = 0$ as a revolution curve and the generatrix $g(z) = \text{Sin}[z/9] * 12 + 17$.

curve (top left of the figure). Voxels are represented smaller in those pictures to emphasize the topological properties (connectivity and tunnels).

3.5 Extensions

The method we have presented can easily be extended to generate various, more general, types of surfaces. One simple extension is to use the generatrix, not as an homothetic function but as central axis for the revolution curve:

Definition 3.

$$S(g, r) = \{x, y, z \in \mathbb{R}^3, r(x, y - g(z)) = 0\}$$

In an other example, we combine a homothetic function $h(z)$ and two functions for translations $t(z), u(z)$ of the center of the revolution curve:

Definition 4.

$$S(h, t, r) = \left\{ x, y, z \in \mathbb{R}^3, r \left(\frac{x}{h(z)} - u(z), \frac{y}{h(z)} - t(z) \right) = 0 \right\}$$

On the left of figure 6 we can see an example where a sinusoid generatrix is considered as a translation function for the revolutions function (here the unit circle). On the right of figure 6, the digital surface is obtained with an homothetic factor $h(z) = |z|/5$, two translation functions $t(z) = \sin(z/5)$ and $u(z) = \cos(z/5)$ and the unit circle $r(x, y) = x^2 + y^2 - 1 = 0$ as a revolution function. The implicit equation of the surface is the one of definition 4.

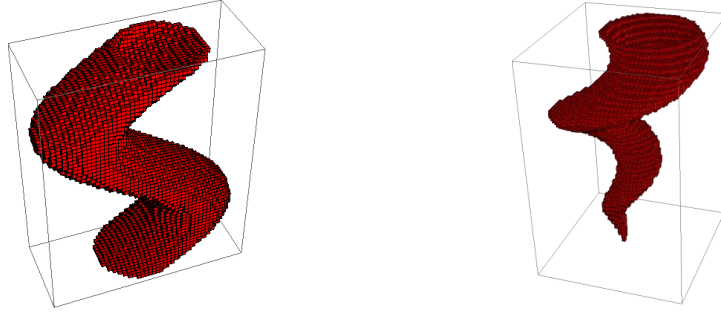


Fig. 6. Two examples of some other surface we can build with our algorithm.

3.6 Limitations

The main limitation of this digital revolution surface generation method is that the analytical digitization can miss some points. Figure 7 (left) shows that a curve can cross a voxel but the vertices of the flake are on only one side of the curve. The voxel is therefore wrongly discarded from the digitization result and disconnections appear. This problem is classically dealt with interval arithmetics [18]. The problem with an interval arithmetic approach is that it is easy to use to replace \mathcal{A}_0 but more complicated for both other analytical digitizations. This could be an interesting problems for the the future.

Figure 7 presents an example where this limitation can be seen: there are some missing voxels in the surface. Those missing voxels correspond to the points where the revolution curve (middle of figure 7) crosses itself.

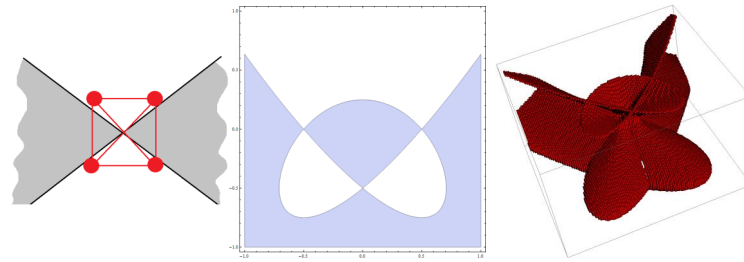


Fig. 7. A_2 -digitization with $r(x, y) = 16y^3 + 12y^2 - (4x^2 - 1)^2 = 0$ as a revolution curve and the generatrix $g(z) = \text{Sin}[z/9] * 12 + 17$. Visible holes in the surface.

4 Conclusion

In this paper, we have presented a very simple generation algorithm for digital surfaces of revolution. We have proposed some extensions that correspond to more general surfaces such as swept surfaces. We need now to investigate that and see if it is possible to create all/most types of digital swept surfaces. The algorithm leads to topologically controlled digital surfaces (0, 1 or 2 connected) upto some special cases where the surface might have unexpected holes. Let us note that this problem seems to appear quite rarely. We had to run a lot of tests in order to get such a figure. Compared to previous methods [4, 5], our method appears to be simpler and more general. Our perspectives are firstly to combine a parametric function as generatrix with implicit revolution curves, the control of the topology in this case could be ensured by the implicit expression of the surface. Secondly, we want to investigate the digitization of full parametric surfaces to allow a hand-drawn generatrix and hand-drawn revolution curve. This should solve some of the problems we have with some configurations where we had holes in the surfaces. The problem in figure 7, for instance, would not occur anymore. We already have some interesting results (see figure 8 for an example) which need to be consolidated.

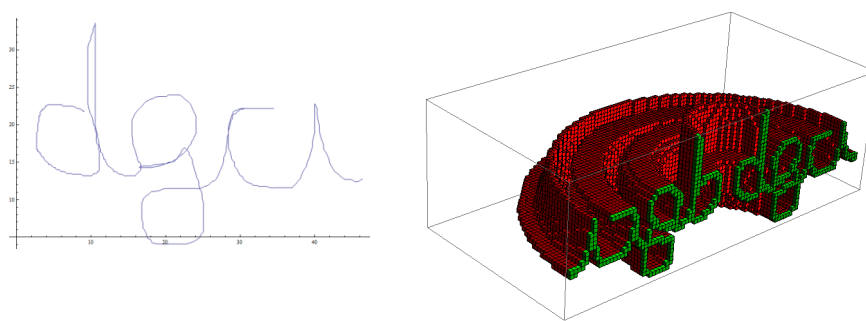


Fig. 8. Hand-drawn generatrix with an implicit circle as revolution curve.

Acknowledgement: This work has been supported by the CPER 2015-2020, NUMERIC Program and FEDER-FSE Project MODEGA of the Poitou-Charentes Region, France.

References

1. Salomon, D.: Curves and surfaces for computer graphics. Springer (2006)
2. Stolte, N., Kaufman, A.E.: Novel techniques for robust voxelization and visualization of implicit surfaces. *Graphical Models* **63**(6) (2001) 387–412
3. Yongsheng, L., Stolte, N.: Robust voxelization based ray tracing of implicit surfaces. In: Proc. 6th Sixth IASTED Honolulu, Hawaii (USA). (2003) 177–180
4. Bhowmick, P., Bera, S., Bhattacharya, B.B.: Digital circularity and its applications. In Wiederhold, P., Barneva, R.P., eds.: *Combinatorial Image Analysis, 13th International Workshop, IWCIA Playa del Carmen, Mexico, November 24-27, 2009. Proceedings.* Volume 5852 of *Lecture Notes in Computer Science.*, Springer (2009) 1–15
5. Kumar, G., Sharma, N.K., Bhowmick, P.: Wheel-throwing in digital space using number-theoretic approach. *IJART* **4**(2) (2011) 196–215
6. Andres, E.: Discrete circles, rings and spheres. *Computer and Graphics* **18**(5) (1994) 695–706
7. Andres, E., Jacob, M.A.: The discrete analytical hyperspheres. *IEEE Trans. on Vis. and Comp. Graphics* **3**(1) (1997) 75–86
8. Andres, E., Roussillon, T.: Analytical description of digital circles. In: 16th DGCI, Nancy (France). Volume 6607 of *LNCS.*, Springer (2011) 235–246
9. Toutant, J., Andres, E., Largeteau-Skapin, G., Zour, R.: Implicit digital surfaces in arbitrary dimensions. In: 18th DGCI, Siena (Italy). Volume 8668 of *LNCS.*, Springer (2014) 332–343
10. Toutant, J., Andres, E., Roussillon, T.: Digital circles, spheres and hyperspheres: From morphological models to analytical characterizations and topological properties. *Discrete Applied Mathematics* **161**(16-17) (2013) 2662–2677
11. Stelldinger, P., Köthe, U.: Towards a general sampling theory for shape preservation. *Image and Vision Computing* **23**(2) (2005) 237–248
12. Breton, R., Sivignon, I., Dupont, F., Andres, E.: Towards an invertible euclidean reconstruction of a discrete object. In: 11th DGCI Naples (Italy). Volume 2886 of *LNCS.*, Springer (2003) 246–256
13. Sivignon, I., Breton, R., Dupont, F., Andres, E.: Discrete analytical curve reconstruction without patches. *Image and Vision Computing* **23**(2) (2005) 191–202
14. Cohen-Or, D., Kaufman, A.E.: Fundamentals of surface voxelization. *CVGIP* **57**(6) (1995) 453–461
15. Andres, E., Acharya, R., Sibata, C.: The supercover 3d polygon. In: 6th DGCI, Lyon (France). Volume 1176 of *LNCS.*, Springer (1996) 237–242
16. Andres, E.: Defining discrete objects for polygonalization: The standard model. In: 10th DGCI, Bordeaux (France). Volume 2301 of *LNCS.*, Springer (2002) 313–325
17. Andres, E.: Discrete linear objects in dimension n: the standard model. *Graphical Models* **65**(1-3) (2003) 92 – 111
18. Duff, T.: Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry. In: *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1992.* (1992) 131–138