# ON ESCHER'S SPIRALS
## Polygonization of 2-manifolds with boundaries

*Alexander A. Pasko*

Shape Modeling Lab., Department of Computer Software, The University of Aizu,
Aizu-Wakamatsu City, Fukushima, Japan 965
pasko@u-aizu.ac.jp
http://www.u-aizu.ac.jp/public/www/labs/sw-sm/pasko

**ABSTRACT**

An algorithm of polygonization of trimmed implicit surfaces yielding surface sheets is presented. These two-dimensional manifolds with boundaries result from the set-theoretic difference of an implicit surface and a solid. The algorithm generates the polygonal approximation of the trimmed surface with the mesh adaptation to the manifold boundary.

### 1. INTRODUCTION

This work was inspired by art works of M.C. Escher, namely "Sphere Spirals" (1958) and "Bond of Union" (1956), showing spiral shaped surface sheets cut of a sphere and human head surfaces. These art works raise two questions:
1) How does one define a geometric model for a surface sheet of this type?
2) How does one visualize such geometric model?

A surface sheet can be mathematically defined as a two-dimensional manifold with boundary (or simply a 2-manifold). Modeling and visualization of the above mentioned spiral type 2-manifolds using parametric surfaces seems to be a difficult task. The alternative is to use isosurfaces of functions of three variables (so-called implicit surfaces). A 2-manifold can be represented as a set-theoretic difference between some initial carrier implicit surface and a trimming solid (see Fig.1 for modeling Escher's "Sphere Spirals"). In this paper we discuss problems of defining carrier surfaces and trimming solids, and polygonization of the trimmed surface with the mesh adaptation to the surface-surface intersection curves composing the manifold boundary.

### 2. OTHER WORKS

If we define the initial surface $A$ as $f_A\,(x,y,z) = 0$ and the trimming solid $B$ as $f_B\,(x,y,z) \geq 0$, the trimmed surface $T = A \setminus B$ can be defined as $f_T\,(x,y,z) = 0$ with

$$f_T = -f_A^{\,2} \,\&\, (-f_B),$$

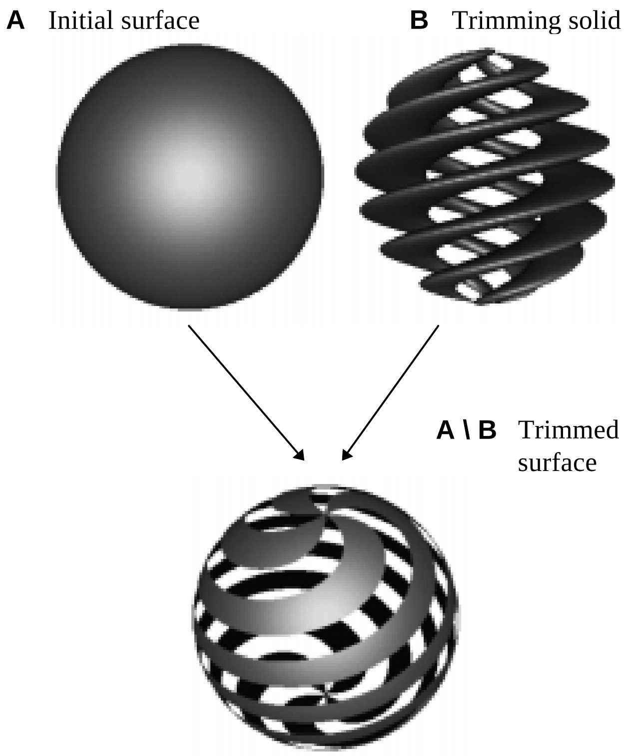



Figure 1. "Sphere Spirals": 2-manifold with boundary definition as a set-theoretic difference between an initial surface A and a trimming solid B.

where symbol & denotes $min(f_1,\ f_2)$ or some other function corresponding to the set-theoretic intersection. The idea here is to represent the surface as a solid with $-f_A^{\,2} \geq 0$ and further to apply set-theoretic operations to it. This was proposed by Rvachev [4]. A similar approach is used in the modern solid modeller Svlis [3]. The serious disadvantage of this approach is that it is not possible to distinguish two sides of the initial surface represented as $-f_A^{\,2} \geq 0$. Therefore, it is not possible to apply conventional polygonization algorithms based on inside-outside point classification [2].

Bloomenthal and Ferguson [1] proposed a general polygonization algorithm for non-manifold surfaces. The algorithm can polygonize both 2-manifolds with boundaries and non-manifold surfaces. Because of the algorithm complexity, it is difficult to implement its adaptive version. A more simple algorithm processing only 2-manifolds with boundaries can be obtained by extending conventional polygonization algorithms.

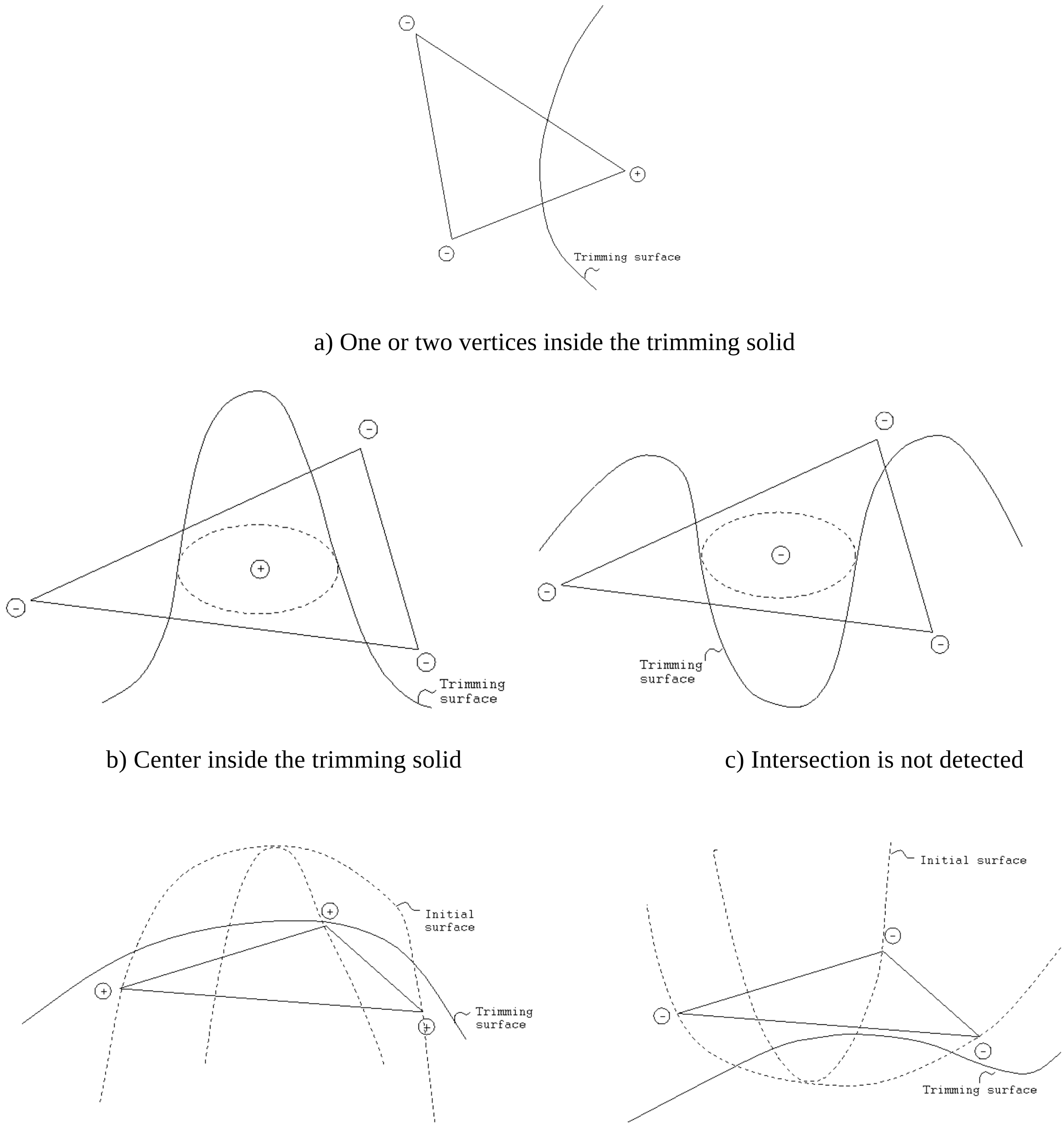
The problem we deal with in this paper is related to the intersection of two implicit surfaces. There are two main numerical approaches to implicit surface-surface intersection :

• Start with some intersection point found analytically or numerically. Trace the intersection curve using some optimization method (for example, a gradient search).

• Approximate both surfaces by polygons and intersect two obtained polyhedrons.

To polygonize a boundary of a CSG solid, Wyvill and van Overveld [7] apply the uniform subdivision algorithm to the resulting surface and then enhance the mesh with an iterative numerical procedure.

Note that the above mentioned methods treat both surfaces equally. On the other hand, one could observe in Fig. 1 that for a simple initial surface the function defining a trimming solid can be very complex. It means that evaluation times for these two functions can differ drastically. Therefore, an algorithm aiming to decrease the number of evaluations of the more complex function can substantially decrease the overall computation time.




a) One or two vertices inside the trimming solid

b) Center inside the trimming solid

c) Intersection is not detected

d) Triangle inside and near the trimming surface

e) Triangle outside and near the trimming surface

Figure 2. Adaptation criteria for a polygon of the initial surface mesh.

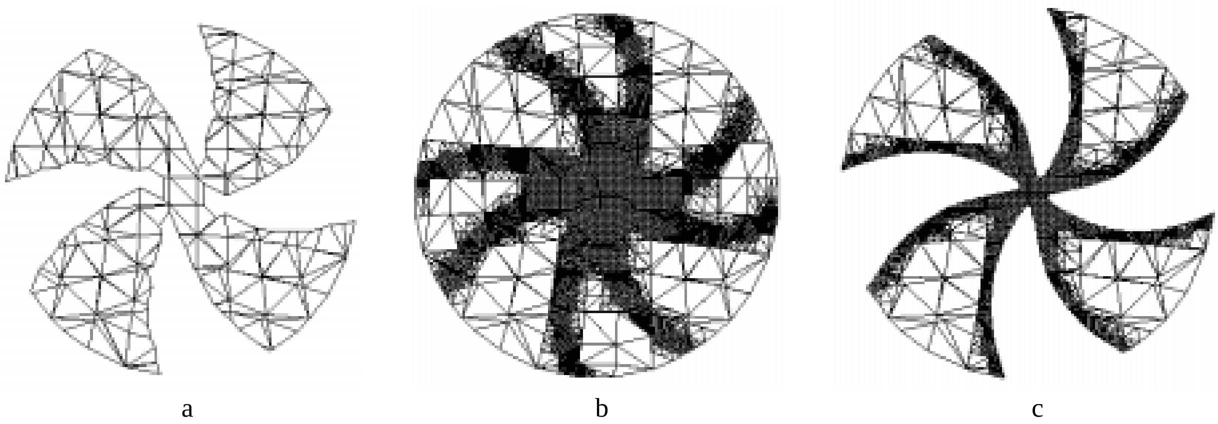


Figure 3. Adaptive trimming of a sphere (top part) by a complex solid:
(a) result of non-adaptive trimming with polygonization on a sparse grid;
(b) mesh obtained by the adaptive polygonization;
(c) result of adaptive trimming.

## 3. TRIMMING ALGORITHM

We propose an algorithm for implicit surface trimming based on the extension of existing implicit surface polygonization algorithms. The idea is to adaptively polygonize the initial surface near its intersection curves with the trimming solid surface and to test the obtained polygons against the trimming solid. The proposed algorithm is based on the implicit surface polygonization method proposed in [5]. In fact, any conventional polygonization algorithm can be extended in this way.

The algorithm includes the following steps:

1. Define the initial surface as $f_A\,(x,y,z) = 0$ and the trimming solid as $f_B\,(x,y,z) \geq 0$.

2. Introduce a sparse spatial grid in $(x,y,z)$ space.

3. Calculate $f_A\,(x,y,z)$ values at the grid points.

4. Obtain the initial triangulation for the surface $f_A = 0$ following a conventional polygonization algorithm providing correct topology with the given sparse grid.

5. For each triangle, calculate $f_B\,(x,y,z)$ values in its vertices and check the following adaptation criteria (see Fig. 2):

- Different signs of the function $f_B$ values in the vertices: the triangle intersects the trimming surface (Fig. 2a);
- Evaluate $f_B$ in the barycenter of the triangle. If the sign is different from the signs in the vertices, the trimming surface penetrates the triangle but all vertices are outside or inside the surface (Fig. 2b). Fig. 2c shows the worst case when the trimming surface intersects the triangle but the sign in the barycenter is the same as signs in the vertices.
- If the absolute value of $f_B$ in a vertex is less than some given ε, the triangle is close to the trimming surface with the possible surface-surface intersection (Fig. 2d,e).

6. If one of the adaptation criteria is satisfied, start recursive subdivision of the triangle in four triangles by introducing new vertices in the middle of its edges. Place newly introduced vertices on the initial surface using a search in the normal direction. If a triangle adjacent to an edge is not subdivided, an additional triangle has to be inserted to prevent cracks. Repeat the previous step for the four new triangles.

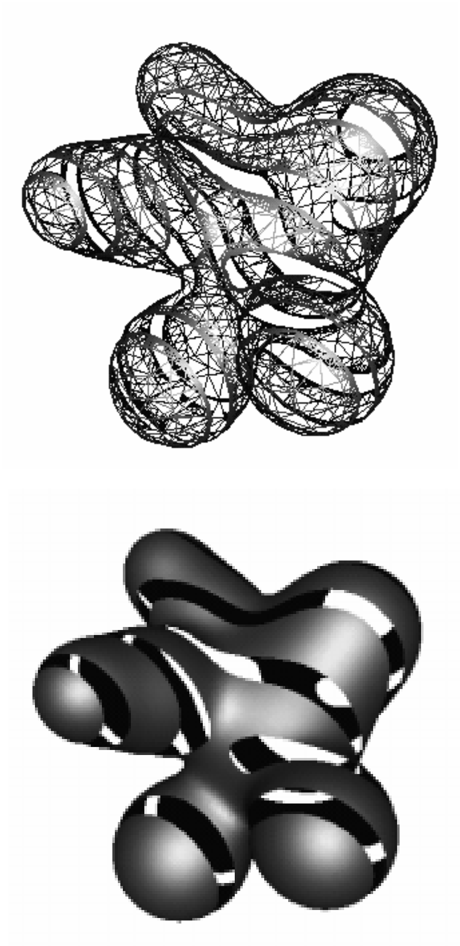
7. If no one of the adaptation criteria is satisfied for the current triangle or the given level of subdivision is achieved, classify the triangle against the trimming solid. Check the values of the function $f_B$ in the vertices. Three cases are possible:

1) Three positive values. The triangle is completely inside the trimming solid and is not included in the resulting mesh.

2) Three negative values. The triangle is completely outside the trimming solid and is included in the resulting mesh.

3) Different signs. The endpoints of the intersection line segment can be found by the linear interpolation along the corresponding edges. The triangle is split along this segment and the part lying outside the trimming solid is included in the resulting mesh.

Note that the proposed algorithm starts with the sparse mesh and then invokes its adaptation in the neighborhood of the intersection curves. Moreover, the function $f_B$ is evaluated only in the vertices of the initial surface mesh but not in all 3D grid points. This helps to decrease the number of time-consuming function evaluations while providing required accuracy of the 2-manifold boundary extraction.



a

b

Figure 4. “Blobby Spiral”: an adaptive polygonal mesh (a) and a blobby object surface (b) trimmed by a solid.

## 4. EXPERIMENTS

The trimmed sphere shown in Fig. 1 was polygonized with the proposed algorithm. The trimming solid was modeled using sweeping, offsetting and union operations [6] (see Fig.5). Its top and bottom critical points are placed exactly on the initial sphere with the trimming surface tangent to the sphere in the critical points. This is necessary to check the proposed adaptive polygonization criteria. Fig. 3 illustrates the difference between non-adaptive and adaptive polygonization near the top critical point on the sparse grid with $13\times13\times9$ nodes and four recursion levels of adaptation. The proposed polygonization algorithm can be applied to any initial surface and a trimming solid defined by continuous real functions. Fig. 4 shows an adaptive polygonal mesh and a trimmed surface of a blobby object. Its adaptive polygonization on the $20\times20\times20$ initial grid took 40 seconds on a SG Indigo[2] workstation. Its non-adaptive polygonization on the corresponding $128\times128\times128$ grid takes about 25 minutes.

```
z2 = z*z;
R = sqrt(100.-z2);
x0 = R*cos(0.5*z+phi);
y0 = R*sin(0.5*z+phi);
xt = x-x0;
yt = y-y0;
r = 2.- z2*0.02;
ftrim = r*r - xt*xt -yt*yt;
offset = 10 - z2*0.1;
ftrim = ftrim+offset;
```

Figure 5. Function *ftrim(x, y, z)* defining a component of the trimming solid for the “Sphere Spirals” (Fig. 1); $x, y, z \in [-10, 10]$, and *phi* defines a phase of the spiral.

## 5. CONCLUSION

A new algorithm of polygonization of two-dimensional manifolds with boundaries is proposed. The algorithm extends conventional polygonization algorithms by including a trimming solid and adapting the polygonal mesh to the 2-manifold boundary. The proposed adaptive solution significantly accelerates polygonization. Depending on the initial surface and the trimming solid complexity, we obtained 20 to 70 times speed-up if compared with the non-adaptive algorithm.

## ACKNOWLEDGEMENTS

The author would like to sincerely thank Professor Vladimir Savchenko for fruitful discussions. Special thanks go to Galina Pasko for the proposed example.

## REFERENCES

[1] Bloomenthal, J., Ferguson, K., *Polygonization of non-manifold implicit surfaces*, in: *SIGGRAPH’95, Computer Graphics Proceedings* (1995) 309-316.

[2] Bloomenthal, J. et al., *Introduction to Implicit Surfaces*, Morgan Kaufmann, 1997.

[3] Bowyer, A., *SVLIS Set-Theoretic Kernel Modeller. Introduction and User Manual*, Information Geometers, Winchester, UK, 1994.

[4] Rvachev, V.L., *Theory of R-functions and Some Applications*, Naukova Dumka, Kiev, 1987 (in Russian).

[5] Pasko, A.A., Pilyugin, V.V., Pokrovskiy V.V., *Geometric modeling in the analysis of trivariate functions*, Computers and Graphics 12 (1988) 457-465.

[6] Pasko, A., Adzhiev, V., Sourin, A., Savchenko, V., *Function representation in geometric modeling: concepts, implementation and applications*, The Visual Computer 11 (1995) 429-446.
http://www.u-aizu.ac.jp/public/www/labs/sw-sm/FrepWWW/F-rep.html

[7] Wyvill, B., van Overveld, K., *Polygonization of implicit surfaces with Constructive Solid Geometry*, International Journal of Shape Modeling 2 (1997) 257-273.