

Implicit Surface Modeling using Supershapes and R-functions

Yohan D. Fougerolle^{1,2}, Andrei Gribok¹, Sebti Fofou², Frédéric Truchetet², and Mongi A. Abidi¹

1. IRIS Lab., University of Tennessee, Knoxville, TN, USA

2. Le2i Lab., UMR CNRS 5158, Université de Bourgogne, Le Creusot, FRANCE

E-mail: {yfougero, agribok, abidi}@utk.edu, {Sebti.Fofou, Frederic.Truchetet}@u-bourgogne.fr

Abstract

We propose a method to represent solid objects as multiple combinations of globally deformed supershapes. Our approach handles primitives that have both parametric and implicit representations. An implicit equation with guaranteed differential properties is obtained by combinations of the primitives' implicit representations using R-functions. The surface corresponding to the zero-set of the implicit equation is efficiently and directly polygonized using the primitives' parametric forms. We illustrate our approach by representing complex models composed of several hundreds of primitives.

Keywords: Solid modeling, Implicit surfaces, Superquadrics and supershapes

1. Introduction

Among the various methods to represent objects, a natural idea is to express complex objects as combinations of simpler ones, which leads to the notion of solid modeling. The compactness of such representation is an important factor: sophisticated implicit functions often require few coefficients whereas parametric surfaces need complex control meshes and need larger data storage. Compact primitives such as quadrics have been generalized to superquadrics, introduced by A. H. Barr [1, 2]. Superquadrics are a special case of the supershapes, proposed by Gielis et al. [4, 5], that have the advantage of representing polygons with various symmetries.

We propose a CSG-based method to represent solid objects using R-functions and supershapes. The parametric definition of each primitive is used to efficiently and accurately generate the resulting mesh: the vertices of the final mesh lie exactly on the surface and sharp edges are preserved. The contributions of this paper are an extension of the literature in solid modeling by a novel combination of supershapes and R-functions applied to a CSG framework, where the resulting implicit surface can be directly and efficiently polygonized. The rest of

paper is organized as follows: the second section briefly presents most common R-functions and their differential properties. The third section deals with supershapes and their implicit and parametric representations. The fourth section presents the algorithm to generate the surface of complex objects from a CSG tree of globally deformed supershapes with sharp edges before our conclusions and future work.

2. Implicit modeling and R-functions

Various geometrically continuous blending functions have been proposed by Ricci [7], Pasko [6], or Rvachev [8]. An English tutorial on R-functions has been proposed by Shapiro [9]. Depending on their differential properties [10], different R-functions may be used, namely R_α , R_0^m and R_p .

$$R_\alpha : \frac{1}{1 + \alpha} \left(x + y \pm \sqrt{x^2 + y^2 - 2\alpha xy} \right), \quad (1)$$

where $\alpha(f_1, f_2)$ is an arbitrary symmetric function such that $-1 < \alpha(f_1, f_2) \leq 1$. R_0^m is defined as

$$R_0^m : \left(x + y \pm \sqrt{x^2 + y^2} \right) (x^2 + y^2)^{\frac{m}{2}}, \quad (2)$$

where m is any even positive integer. R_p -function is defined as

$$R_p : x + y \pm (x^p + y^p)^{\frac{1}{p}}, \quad (3)$$

for any even positive integer p .

3. Supershapes: parametric and implicit formulations

Supershapes have been recently presented by Gielis [4, 5] as an extension of superquadrics. Deriving from the superellipse representation, a term $\frac{m\theta}{4}$, $m \in \mathbb{R}^+$, is introduced to allow a rational or irrational number of symmetry and three shape coefficients are considered. The radius r of a superpolygon

is defined by

$$r(\theta) = \frac{1}{n_1 \sqrt{\left|\frac{1}{a} \cos\left(\frac{m\theta}{4}\right)\right|^{n_2} + \left|\frac{1}{b} \sin\left(\frac{m\theta}{4}\right)\right|^{n_3}}}, \quad (4)$$

with a, b , and $n_i \in \mathbb{R}^+$, and $m \in \mathbb{R}_*^+$. Parameters $a > 0$ and $b > 0$ control the size of the polygon, $m \geq 0$ defines the number of symmetry axis and can also be seen as the number of sectors in which the plane is folded. When m is a natural number, non-self-intersecting closed curves are obtained. A condensed parametric equation of supershapes can be written as

$$\begin{pmatrix} x(\theta, \phi) \\ y(\theta, \phi) \\ z(\theta, \phi) \end{pmatrix} = \begin{pmatrix} r_1(\theta)r_2(\phi) \cos \theta \cos \phi \\ r_1(\theta)r_2(\phi) \sin \theta \cos \phi \\ r_2(\phi) \sin \phi \end{pmatrix}, \quad (5)$$

with $-\pi \leq \theta < \pi$ and $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$. A unit supershape ($a = b = 1$) is defined by 8 shape parameters noted $\{m, n_1, n_2, n_3, M, N_1, N_2, N_3\}$, where n_i and N_i are used in $r_1(\theta)$ and $r_2(\phi)$ respectively. In [4], two similar distance functions are proposed for supershapes: the authors' idea is to project a considered point onto the two orthogonal generating planes of the supershape to independently evaluate two distances. In [3], we proposed Eq. 6 as a supershape implicit function.

$$F(x, y, z) = 1 - \frac{x^2 + y^2 + r_1^2(\theta)z^2}{r_1^2(\theta)r_2^2(\phi)}. \quad (6)$$

We introduce another formulation based on the notion of radial distance, introduced in Eq.7 by expressing a function of the ratio $t = \frac{\|\vec{OI}\|}{\|\vec{OP}\|}$, where I is the intersection of the supershape surface and the half line $[OP)$.

$$\begin{aligned} F(x, y, z) &= 1 - \frac{\|\vec{OP}\|}{\|\vec{OI}\|} = 1 - t \\ &= 1 - \frac{1}{r_2(\phi)} \sqrt{\frac{x^2 + y^2 + z^2}{\cos^2 \phi (r_1^2(\theta) - 1) + 1}}. \end{aligned} \quad (7)$$

It is important to notice that θ and ϕ do not correspond to the spherical angles, actually, it is easy to see that spherical and supershape angles are equal only for $r_1(\theta) = 1$. For a point $P(x, y, z)$, angles θ and ϕ are defined by

$$\begin{cases} \theta = \theta(x, y) & = \arctan\left(\frac{y}{x}\right) \\ \phi = \phi(x, y, z, n_1, n_2, n_3) & = \arctan\left(\frac{zr_1(\theta)\sin(\theta)}{y}\right) \\ & = \arctan\left(\frac{zr_1(\theta)\cos(\theta)}{x}\right). \end{cases} \quad (8)$$

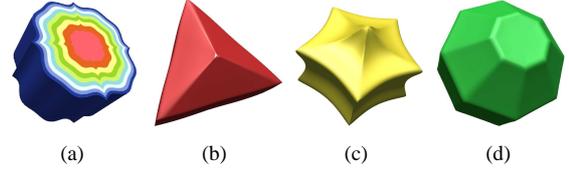


Figure 1: a) Intensity of the scalar field generated for a planar section $z = \text{const}$. b), c), and d) Examples of supershapes

Fig. 1(a) shows the intensity of the scalar field for a plane $z = \text{const}$ using Eq. 6 and examples of supershapes with various sets of shape parameters. To increase the range of shapes that can be modelled, we consider four global transformations (scaling, tapering, twisting, and bending) as in [2], that are not only applied to single primitives but to complete subtrees as proposed in [11].

4. Generating the surface of the CSG tree

To create a mesh accurately representing the surface of the object, we isolate the parts of the supershapes that must be kept and merge them along the approximations of the intersection curves. The process can be split in two steps: the initialization, where the primitives are polygonized, and the node evaluation, presented in algorithm 1, where the surface is built and refined.

Algorithm 1 Recursive node evaluation

Input: two CSG subtrees

1. Inside/outside evaluation using the subtrees' implicit functions

2. Create intersection curves

for all Intersecting faces **do**

Create intersection points I such as $F(I) < \varepsilon$

Build intersection curves

end for

3. Re-sample intersections curve to assert $d(I_n, I_{n+1}) < \delta$, where I_n and I_{n+1} are consecutive intersection points

4. Split faces

5. Merge intersection curves

6. Transfer vertices and faces to parent node

Output: Closed mesh transferred to the parent node

The sign of the implicit function of each subtree is used in algorithm 1 as a characteristic function to determine if a vertex is lying inside or outside the volume defined by the other subtree. Faces are considered completely inside/outside if their vertices are completely inside/outside the resulting object. By

$F(I) < \varepsilon$, we mean that we insert an approximation of an intersection point I when the implicit function of the other subtree is inferior to ε . Since I is created using a primitive parametric form, $f_{primitive}(I) = 0$, which implies that the resulting R-function is also null. So, even if I is only an approximation of a theoretical intersection point, it still lies exactly on the resulting surface.

Once the inside/outside evaluation is performed for vertices, two types of faces have to be considered: the faces that are completely inside (or outside) the object, and the faces crossed by intersection curves. The first ones are directly transmitted to the parent node in the CSG tree depending on the Boolean operation performed. The second ones must be split along the intersection curves. During this step, we deal with two different issues. The first one concerns the scales of the objects, and the second the accuracy of the approximation of the intersection curves. We introduce two user defined parameters ε and δ to control the accuracy and the density of vertices representing the intersection curves. Parameter δ controls the maximum distance between two consecutive vertices along the approximation of the intersection curve. Its role is to interface the gaps in term of sampling or scale that may arise when evaluating large CSG trees. Parameter δ is adjusted in function of the smallest edge length of the two objects that are combined to avoid cracks and holes when combining primitives with very different scales or samplings. The last step of the algorithm consists in merging the parts from both objects and closing the resulting mesh. We applied our method to represent a complex CSG tree, inspired from an existing CAD model presented in Fig. 2(a). The CSG tree is composed of 235 supershapes and 238 Boolean operations.

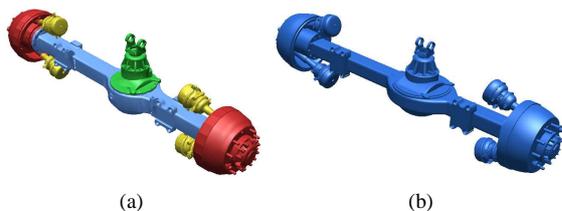


Figure 2: Axle Model. a) Inspiring mesh b) Super-shape representation.

5. Conclusions and future work

In this paper, we have proposed an algorithm to accurately polygonize the resulting surface of an implicit function of a solid defined as multiple Boolean operations between hierarchically deformed super-

shapes and a new implicit function for supershapes. Our framework combines the advantages of using R-functions with guaranteed differential properties, with powerful, versatile, and compact primitives. Our method preserves interesting properties of implicit and parametric techniques: R-function theory allows us to efficiently evaluate the parts of primitives to be kept, and the parametric representation of the primitives is used to accurately refine the resulting mesh around the intersection curves within the desired accuracy. Our future work includes the application of this framework to reconstruct supershapes from 3D real data to apply our method to reverse engineering and computer vision.

6. References

- [1] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):481–484, 1981.
- [2] A. H. Barr. Global and local deformation of solid primitives. *Computer Graphics*, 18(3):21–30, 1984.
- [3] Y. D. Fougerolle, A. Gribok, S. Fougou, F. Truchetet, and M. A. Abidi. Boolean operations with implicit and parametric representation of primitives using R-functions. *IEEE Transaction on Visualization and Computer Graphics*, 11(5):529–539, 2005.
- [4] J. Gielis, B. Beirinckx, and E. Bastiaens. Superquadrics with rational and irrational symmetry, 2003. Symposium on Solid Modeling and Applications.
- [5] J. Gielis. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany*, 90:333–338, 2003.
- [6] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.
- [7] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16(2):157–160, 1972.
- [8] V. L. Rvachev. *Geometric Applications of Logic Algebra*. Naukova Dumka, 1967. In Russian.
- [9] V. Shapiro. Theory of R-functions and applications: A primer. Technical Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY, 1991.
- [10] V. Shapiro and I. Tsukanov. Implicit functions with guaranteed differential properties. In *Symposium on Solid Modeling and Applications*, pages 258–269, 1999.
- [11] B. Wyvill, A. Guy, and E. Galin. Extending the CSG tree - warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum*, 18(2):149–158, 1999.