Mar. 2006, Vol.21, No.2, pp.238–243 J. Comput. Sci. & Technol.

Radial Supershapes for Solid Modeling*

Yohan D. Fougerolle[1], Andrei Gribok[2], Sebti Foufou[1], Frédéric Truchetet[1], and Mongi A. Abidi[2]

[1] *Le2i Lab. UMR CNRS 5158, Université de Bourgogne, 12 rue de la fonderie, 71200 Le Creusot, France*

[2] *IRIS Lab., The University of Tennessee, Ferris Hall, Knoxville, 37996 TN, U.S.A.*

E-mail: {y.fougerolle, s.foufou, f.truchetet}@u-bourgogne.fr; {agribok, abidi}@utk.edu

Revised December 12, 2005.

Abstract In the previous work, an efficient method has been proposed to represent solid objects as multiple combinations of globally deformed supershapes. In this paper, this framework is applied with a new supershape implicit function that is based on the notion of radial distance and results are presented on realistic models composed of hundreds of hierarchically globally deformed supershapes. An implicit equation with guaranteed differential properties is obtained by simple combinations of the primitives' implicit representations using R-function theory. The surface corresponding to the zero-set of the implicit equation is efficiently and directly polygonized using the primitives' parametric forms. Moreover, hierarchical global deformations are considered to increase the range of shapes that can be modeled. The potential of the approach is illustrated by representing complex models composed of several hundreds of primitives inspired from CAD models of mechanical parts.

Keywords implicit surface, solid modeling, superquadrics, supershape

1 Introduction

Among various existing 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[1−4]. Two major approaches, with their inherent strengths and weaknesses, are known as constructive solid geometry (CSG) and boundary representation (B-Rep). Constructive solid geometry allows simple, natural, and fast combinations of primitives by Boolean operations. Unfortunately, obtaining the resulting mesh representing the zero-set of the implicit function requires either to explicitly solve the implicit equation, or to use polygonization algorithms such as Marching Cubes. Moreover, such algorithms can only approximate the surface, and in the case of implicit functions built from a CSG tree, sharp edges have to be treated specifically[5,6]. Boolean operations are slower and more difficult to perform on B-reps using parametric surfaces, such as NURBS[7,8], or subdivision surfaces[9], due to the lack of characteristic function to directly evaluate the interiority of a point to an object. In counterpart, the vertices, normals, and higher order derivatives can be directly and, most importantly, exactly computed. Other methods, known as hybrid methods, try to combine different representations or apply a same representation to various kinds of primitives. For example, Adzhiev *et al.*[10] combine volume representation by voxels and real continuous functions; Allègre *et al.*[11] apply the CSG framework to skeletal implicit functions and polygonal meshes.

The compactness of the representation is also an important factor to be considered: sophisticated implicit functions often require few coefficients whereas parametric surfaces need complex control meshes that may need larger data storage but allow easier editions. Compact primitives such as quadric have been generalized to superquadrics, introduced by A. H. Barr[12,13], and have been used for various applications, both in computer graphics and in computer vision[14−16]. Initially aiming at the representation of natural shapes, a recent extension of superquadrics, namely the supershapes, has been proposed by Gielis *et al.*[17,18]. Supershapes have two major advantages: they can simply represent regular polygons and natural shapes with various symmetries and they have both a parametric and an implicit representation[19]. Depending on the branch of R-function used, differential properties for the resulting implicit function are guaranteed. 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 the new implicit function for supershapes, that is based on the notion of radial distance and its application to the representation of models that are composed of hundreds of hierarchically globally deformed supershapes. The rest of this paper is organized as follows. Section 2 briefly presents most common R-functions and their differential properties. Section 3 deals with supershapes and their implicit and parametric representations. Section 4 presents in details the algorithm to generate the surface of complex objects from a CSG tree of globally deformed supershapes with sharp edges. Results are presented in Section 5 before our conclusions and future work.

2 Implicit Modeling and R-Functions

The primitives we consider have both a parametric and an implicit representation, which allows us to take advantage of implicit modeling techniques. $\min(x, y)$ and $\max(x, y)$ functions are among the simplest functions known to blend objects. Unfortunately, these functions are not differentiable along the line $x = y$. Geometrically continuous blending functions, such as the ones introduced by Ricci[20], Blinn[21], or Pasko[22], can be used

*A preliminary version of this paper appeared in Proc. Pacific Graphics 2005, Macau.

to generate smooth objects. In this category also fall R-functions, introduced by Rvachev[23]. An English tutorial has been proposed by Shapiro[24]. We will denote R-conjunction and R-disjunction with the same notation (symbol $\pm$), with R-conjunction obtained by considering $-$ and R-disjunction by considering $+$. R-negation is the opposite sign function $\overline{x} \equiv -x$. Depending on their differential properties[25], different R-functions may be used, namely R_α, R_0^m and R_p. An illustration of R-disjunction using R_α, R_0^m and R_p is presented in Fig.1.

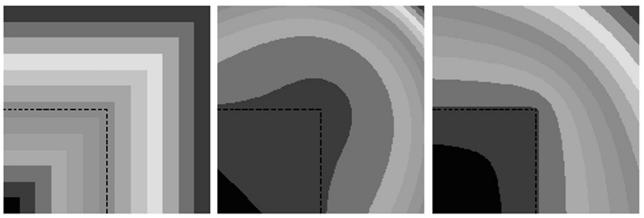

Fig.1. Intensity of the scalar field of the R-disjunction of two variables $x \in [-1,1]$ and $y \in [-1,1]$. On the dashed line, the zero-set of the R-function. Left: $\max(x,y)$, middle: $R_0^m(x,y)$ with $m=2$, and right: $R_p(x,y)$ with $p=2$.

2.1 Properties of R_α

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

where $\alpha(f_1,f_2)$ is an arbitrary symmetric function such that $-1 < \alpha(f_1,f_2) \leqslant 1$. Setting α to 1 leads to the simplest and most popular R-functions: $\min(x,y)$ and $\max(x,y)$.

2.2 Properties of R_0^m and R_p

To solve the loss of differentiability along the line $x=y$, two other functions R_0^m and R_p are proposed.

$$R_0^m : (x+y \pm \sqrt{x^2+y^2})(x^2+y^2)^{\frac{m}{2}} \qquad (2)$$

where m is any even positive integer. Shapiro shows in [25] that R_0^m is m times differentiable everywhere, including the corner point $x=y=0$. All partial derivatives are identically zero at corner points, but unfortunately, R_0^m is not normalized. R_p-function achieves normalization and is differentiable everywhere but at the corner point $x=y=0$.

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

for any even positive integer p.

3 Supershapes: Parametric and Implicit Formulations

Supershapes have been recently presented by Gielis[17,18] as an extension of superquadrics. Gielis uses the terminology superpolygon to describe 2-D supershapes, the term supershape is used for 3-D. Deriving from the superellipse representation, a term $\frac{m\phi}{4}$, $m \in \mathbb{R}^+$, is introduced to allow a rational or irrational number of symmetry and three shape coefficients n_1, n_2, and n_3 are considered. In polar coordinates, the radius r of a superpolygon is defined by

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

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, $m \geqslant 0$ defines the number of symmetry axis and can also be taken as the number of sectors in which the plane is folded. For $m=4$ and $n_2 = n_3$, the original superellipses are obtained. When m is a natural number, non-self-intersecting closed curves are obtained, and regular polygons can be generated by setting the shape coefficients to specific values as shown in [17]. Extension to 3D is performed by the spherical product of two 2D superpolygons as done for superquadrics in [13]. In this paper, we are interested in solid objects and non self-intersecting surfaces, which implies m has to be an integer. A condensed parametric version can be written as

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

with $-\pi \leqslant \theta < \pi$ and $-\frac{\pi}{2} \leqslant \phi \leqslant \frac{\pi}{2}$. A unit supershape ($a=b=1$) is defined by 6 shape parameters noted as $\{n_1,n_2,n_3,\ N_1,\ N_2,\ N_3\}$, where n_i and N_i are used in $r_1(\theta)$ and $r_2(\phi)$ respectively. In [17], 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. We propose two new implicit functions, the first one being directly deduced from the previous parametric definition of supershape in (5), the second based on the notion of radial distance. The advantage of these formulations is to obtain one signed value for a given point to determine the inside, the outside or the surface of a given supershape. A first implicit formulation for supershape, derived from its parametric formulation in (5), is defined by

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

For the second formulation, we express a function of the ratio $t = \frac{OP}{OI}$ where the point I is the intersection of the supershape surface and the half line $[OP)$. When $t=1$, P lies on the surface of the supershape ($t<1$ means P is inside the supershape, and for $t>1$ P lies outside the supershape). In order to stay consistent with the classical definition of an implicit surface, i.e., the surface corresponds to the zero-set of the implicit function, we translate the result by 1, which leads to

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

$$= 1 - \frac{1}{r_2(\phi)} \sqrt{\frac{x^2 + y^2 + z^2}{\cos^2 \phi (r_1^2(\theta) - 1) + 1}}. \quad (7)$$

It is important to notice that θ and ϕ do not exactly correspond to the classical spherical angles, actually, it is easy to see that spherical and supershape angles are equal only for $r_1(\theta) = 1$. For a point $P(x, y, z)$, expressed in the canonical referential, and a unit supershape ($a = b = 1$ for both generating superpolygons), angles θ and ϕ are defined by

$$\begin{cases} \theta = \arctan\left(\dfrac{y}{x}\right) \\ \phi = \arctan\left(\dfrac{z r_1(\theta) \sin(\theta)}{y}\right) \\ \quad = \arctan\left(\dfrac{z r_1(\theta) \cos(\theta)}{x}\right). \end{cases} \quad (8)$$

Notice angle ϕ is the function of the coordinates of the point P, the angle θ, and the 3 shape coefficients used in r_1. To determine θ and ϕ, we have to solve the equation $I(\theta, \phi) = tP$, i.e., find θ and ϕ such that the points I, P, and O are aligned as illustrated in Fig.2. Among the advantages of the supershapes are their dual parametric and implicit representations. This means we can directly and exactly generate solutions of the implicit equation by using the parametric representation. Since I lies on the surface of a supershape, it verifies (5). We can see that $\frac{P_y}{P_x} = \frac{I_y}{I_x} = \tan\theta$, which leads us to angle θ. Once θ is known, the radius of the first generating superpolygon $r_1(\theta)$ can be computed. Using a similar idea, i.e., expressing quantity $\tan\phi$ from (5), it can be seen that $\frac{P_z r_1(\theta) \sin\theta}{P_y} = \frac{I_z r_1(\theta) \sin\theta}{I_y} = \tan\phi$, which gives us (8). In other words, for a given point $P_i(x, y, z)$, we evaluate the angles θ and ϕ to compute radii r_2 and r_1 that are then used to evaluate equations (6) or (7). The reason why we consider parameter m as an integer and not a rational for generating superpolygons now appears clearly: with m rational the surface would still be closed, but it would self-intersect, which means there would exist many intersections between the surface and the half line $[OP)$. Fig.2 shows the intensity of the scalar field for a plane $z = const$ using the first formulation. Fig.3 shows examples of supershapes with various sets of shape parameters.

We have now defined two implicit functions for unit supershape. To increase the range of shapes that can be modelled, we consider four global transformations (scaling, tapering, twisting, and bending) as illustrated in Fig.4. Furthermore, deformations are not only applied to single primitives but to complete subtrees as proposed by Wyvill in [26]. Without loss of generality, the implicit function of a globally deformed supershape is defined by

$$F(P_{can}) = F_{can}(D^{-1}(P)) \quad (9)$$

where F_{can} represents the implicit function in the canonical referential and D^{-1} the inverse transformation to bring P to P_{can} in the canonical referential. In our case, D contains the usual transformations, such as rotations and translations, and the four deformations mentioned earlier. In the case of a CSG tree, where each node contains its own deformations and transformations, the reverse transformation to evaluate P_{can} is hierarchically evaluated from the root to the current node.

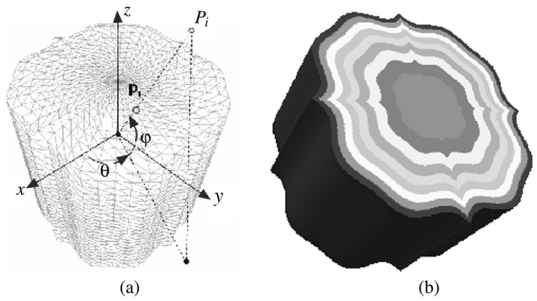

Fig.2. Supershapes. (a) Geometric interpretation. (b) Intensity of the scalar field generated for a planar section $z = const$ using (6).

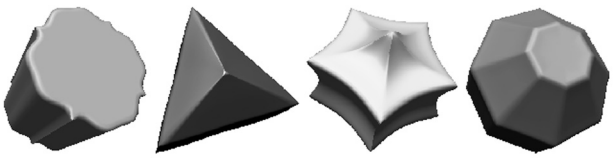

Fig.3. Example of 3D supershapes with m a natural number. From left to right: $m = 3$, $M = 6$; $m = 5$, $M = 6$; $m = 8$, $M = 4$; $m = M = 8$.

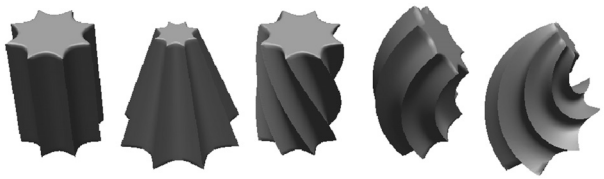

Fig.4. Example of deformations. From left to right: scaled, tapered, twisted, and bent supershape. The last represents the combination of the four deformations.

4 Generation of the Surface of the CSG Tree

We detail in this section the different steps of our algorithm to create the surface of the resulting object. We consider in input of our algorithm a CSG tree of globally deformed supershape. The output includes three components: an implicit function, a union of parametric intervals of the primitives, and a mesh representing the resulting surface. The inspiring CAD model considered to illustrate our algorithm is presented in Fig.9. The decomposition of such model into a CSG tree is not trivial and has been done manually. Our goal in this paper is not to reverse engineer existing complex CAD models, but to show that, even in the case of numerous Boolean operations, sharp edges are preserved using our method. A simpler part, presented in Fig.5, is used for intermediate steps of the algorithm. An illustration of the intensity of the scalar field of the resulting R-function using R_p-function with $p = 2$ and (6) is presented in Fig.11.

Fig.6 presents a simplified CSG Tree of the "Hub" model and the intermediate results during the evaluation

of the resulting surface.

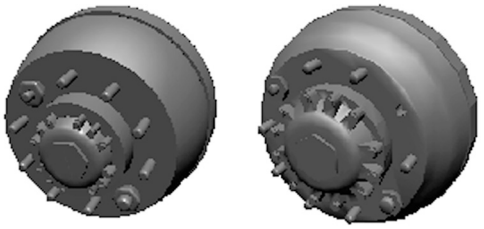

Fig.5. Brake hub model: the corresponding CSG tree is composed of 54 supershapes and 31 Boolean operations.

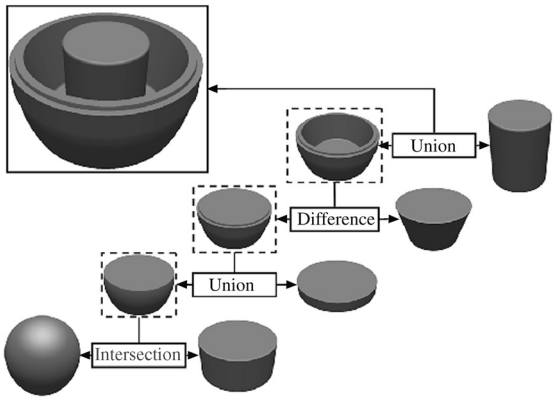

Fig.6. Simplified "Hub" and intermediate results.

Once the CSG tree is defined, an implicit equation for the resulting object can be obtained by combinations of the primitives' implicit equations through R-functions. To create a mesh accurately representing the surface of the resulting object, including the sharp edges, 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 into two simple algorithms: the initialization, where the primitives are polygonized and the deformation hierarchically applied and the node evaluation, where the surface is going to be built and refined. Because of the tree structure, the main algorithm, i.e., the node evaluation, is recursive. The sign of the implicit function of each subtree is used 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. Since vertices P, even approximation of intersection points, are created using a primitive parametric form, $F(P) = 0$, which implies that the resulting R-function is also null.

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 that are crossed by an intersection curve. 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 curve before being transmitted. During this operation, we have to deal with two different issues. The first one concerns the scales of the objects and their samplings, 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, an illustration of their influence on split faces is presented in Fig.7.

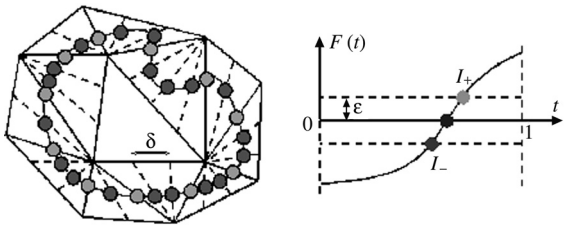

Fig.7. Influence of parameters ε and δ on the intersection curve. Parameter δ controls the sampling along the intersection curve (darker dots).

Since we are not able to directly solve the implicit equation $F = G$, where F and G represent the implicit equation of the two considered CSG trees, we can only approximate the desired solutions. Using the parametric definition of the primitives, the mesh can be refined to generate vertices lying exactly on the resulting surface and that are approximating the intersection curve within an ε accuracy. In practice, we consider $|F(P)| < \varepsilon = 10^{-6}$ sufficiently robust to quickly obtain accurate intersection points. It is also possible to consider a minimum distance along the gradient, or define an attraction force as in [5], because the resulting implicit function can be differentiable everywhere, if R_0^m or R_p-functions are used. This technique has the advantage to be completely independent of the scale of the objects. 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 important 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 different samplings. The last step of the algorithm consists in merging the parts from both objects and closing the resulting mesh. A technique can be found in [5], and we also perform a merging operation between close or

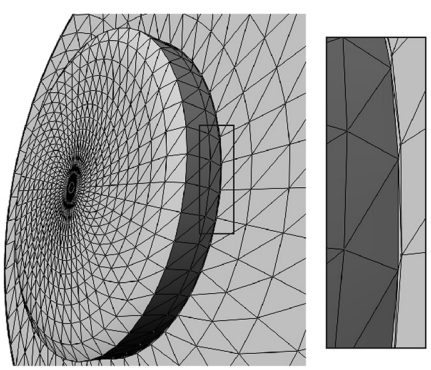

Fig.8. Example of intersection curve after merging.

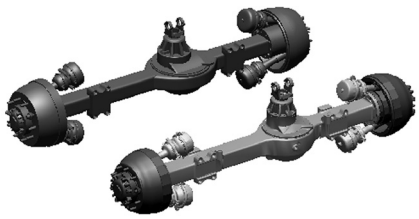

Fig.9. Complete axle mesh and its implicit representation using supershapes and R-functions.

Fig.10. "Shaft": original CAD model and supershape representation.

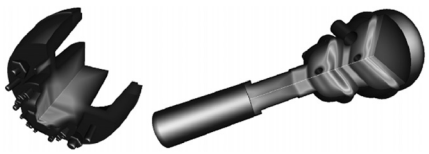

Fig.11. Examples of intensity of the scalar of the resulting R-function for the stabilizer and the brake hub.

identical vertices. An example of merged frontier is shown in Fig.8.

As a remark, computing directly the R-function at the root of the CSG tree is also possible and has been tested. Using this technique avoid all the recursive data transfers from the leaves to the root of the tree. Unfortunately, it has also several drawbacks:

• The simpler the R-function, the smaller the computational error. This becomes crucial during intersection curve approximation.

• It is easier and faster to merge the frontiers of two objects than many frontiers of many objects: by considering binary trees and working recursively, we avoid other problems about the topology of the object near the merged frontiers and keep intermediate meshes closed.

5 Results

We applied our method to represent a complex CSG tree, inspired from an existing CAD model as presented in Fig.9. The complete CSG tree is composed of several hundreds of nodes: 235 supershapes combined through 238 Boolean operations. Using the symmetry properties of the model, only four parts needed to be represented before being copied and pasted to build the final object. Two parts and their implicit representation are presented in Fig.10. Additionally, the intensity of the scalar field of the R-function is presented on two sliced subparts in Fig.11.

6 Conclusions and Future Work

In this paper, we proposed an extension of our algorithm to accurately polygonize the resulting surface of an implicit function describing the volume of a solid that is defined as multiple Boolean operations between hierarchically deformed supershapes. Our framework combines the advantages of using R-functions with guaranteed differential properties to transcribe Boolean predicates into analytical functions, with powerful, versatile, and compact primitives. Our method preserves most of the interesting properties of implicit and parametric techniques: using R-function theory allows us to efficiently evaluate the parts of primitives that must be kept, and the parametric representation of the primitives is used to efficiently refine the resulting mesh around the intersection curves within the desired accuracy. Even if sharp edges cannot be exactly computed, intersection curves can be approximated to an ε-accuracy and the vertices representing intersection points are lying exactly on the zero set of the resulting R-function. Another parameter δ is also used to control the density of points along intersection curves and to avoid cracks and holes when parts of primitives with important sampling or scale disparities are merged. Due to the implicit formulation underlying for every node, the time complexity for node evaluation stays linear in the number of vertices considered which allows to represent complex CSG trees where hundreds of Boolean operations are performed. The implicit function for supershape that is introduced in this paper is based on the notion of radial distance. Being now able to represent any complex object by a CSG tree of primitives, i.e., a single implicit equation, we are going to study the potential solutions to retrieve more accurately the CSG tree structure from a cloud of points. The radial formulation of supershapes also presents interesting isotropy properties that are going to be used to develop surface reconstruction algorithms that are going to be applied to reverse engineering and computer vision.

References

[1] Hoffmann C M. Geometric and Solid Modeling: An Introduction. Morgan Kaufmann Publishers Inc., 1989.

[2] Rossignac J R. Constraints in constructive solid geometry. In *Proc. the 1986 Workshop on Interactive 3D Graphics*, Chapel Hill, North Carolina, USA, 1986, pp.93–110.

[3] Rossignac J R, Requicha A A G. Constructive non-regularized geometry. *Computer-Aided Design*, 1991, 23(1): 21–32.

[4] Rossignac J R, Voelcker H B. Active zones in csg for accelerating boundary evaluation, redundancy elimination, interference detection, and shading algorithms. *ACM Trans. Graphics*, 1989, 8(1): 51–87.

[5] Ohtake Y, Belyaev A, Pasko A. Dynamic mesh optimization for polygonized implicit surfaces with sharp features. *The Visual Computer*, 2003, 19(2-3): 115–126.

[6] B Wyvill, K Van Overveld. Polygonization of implicit surfaces with constructive solid geometry. *J. Shape Modelling*, 1996, 2(4): 257–274.

[7] Krishnan S, Manocha D, Gopi M, Culver T, Keyser J. Boole: A boundary evaluation system for Boolean combinations of sculptured solids. *International Journal of Computational Geometry & Applications*, 2001, (1): 105–144.

[8] Piegl L, Tiller W. The NURBS Book. Second Edition, Springer, 1997.

[9] Biermann H, Kristjansson D, Zorin D. Approximate Boolean operations on free-form solids. In *Proc. SIGGRAPH'01*, Los Angeles, CA, USA, 2001, pp.185–194.

[10] Adzhiev V, Kazakov M, Pasko A, Savchenko V. Hybrid system architecture for volume modeling. *Computers & Graphics*, 2000, 24(1): 194–203.
[11] Allègre R, Barbier A, Galin E, Akkouche S. A hybrid shape representation for free-form modeling. In *Proc. Shape Modeling International*, Genova, Italy, June 7–9, 2004, pp.7–18.
[12] Barr A H. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1981, 1(1): 481–484.
[13] Barr A H. Global and local deformation of solid primitives. *Computer Graphics*, 1984, 18(3): 21–30.
[14] Jaklič A *et al.* Segmentation and Recovery of Superquadrics. Dordrecht: Kluwer Academic Publisher, 2000.
[15] Jaklič A, Solina F. Moments of superellipsoids and their application to range image registration. *IEEE Trans. Systems, Man, and Cybernetics*, 2003, 33(4): 648–657.
[16] Montiel M E , Aguado A S, Zaluska E. Surface subdivision for generating superquadrics. *The Visual Computer*, 1998, 14(1): 1–17.
[17] Gielis J, Beirinckx B, Bastiaens E. Superquadrics with rational and irrational symmetry. In *Proc. 8th ACM Symp. Solid Modeling and Applications,* Seattle, USA, 2003, pp.262–265.
[18] Gielis J. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany*, 2003, 90: 333–338.
[19] Fougerolle Y D, Gribok A, Foufou S, Truchetet F, Abidi M A. Boolean operations with implicit and parametric representation of primitives using R-functions. *IEEE Trans. Visualization and Computer Graphics*, 2005, 11(5): 529–539.
[20] Ricci A. A constructive geometry for computer graphics. *The Computer Journal*, 1972, 16(2): 157–160.
[21] Blinn J F. A generalization of algebraic surface drawing. *ACM Trans. Graphics*, 1982, 1(3): 235–256.
[22] Pasko A, Adzhiev V, Sourin A *et al.* Function representation in geometric modeling: Concepts, implementation and applications. *The Visual Computer*, 1995, 11(8): 429–446.
[23] Rvachev V L. Geometric Applications of Logic Algebra. Naukova Dumka, 1967. (In Russian)
[24] Shapiro V. Theory of R-functions and applications: A primer. Technical Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY, 1991.
[25] Shapiro V, Tsukanov I. Implicit functions with guaranteed differential properties. In *Symposium on Solid Modeling and Applications*, Ann Arbor, Michigan, USA, 1999, pp.258–269.
[26] Wyvill B, Guy A, Galin E. Extending the CSG tree — Warping, blending and Boolean operations in an implicit surface modeling system. *Computer Graphics Forum*, 1999, 18(2): 149–158.

Yohan D. Fougerolle received his M.S. degree in electrical engineering from the University of Burgundy, Dijon, France, in 2002. He is a Ph.D. candidate in electrical and computer engineering at the University of Burgundy, Le Creusot, France and was a visiting research scholar in the Imaging, Robotics, and Intelligent Systems Laboratory, Knoxville, Tennessee from 2002 to 2005. His research interests include computer vision, solid modeling and surface reconstruction.

Andrei Gribok is a research assistant professor in the Department of Computer and Electrical Engineering at the University of Tennessee, Knoxville and also holds a position of an adjunct professor of statistics in the Department of Statistics at the same university. His area of expertise is inverse and ill-posed problems in engineering, statistical learning and model misspecification in statistics. His publications list includes three book chapters and numerous journal and conference papers. Dr. Gribok has a B.S. degree in system science and an M.S. degree in nuclear engineering from the Moscow Institute of Physics and Engineering. He received his Ph.D. degree from the Moscow Institute of Biological Physics in the area of acoustical pattern recognition, artificial intelligence and nondestructive testing. He has 15 years of experience in industry as well as in academia. Dr. Gribok worked as an invited scientist in the Cadarache Nuclear Research Centre in France where his work focused on nuclear power plants monitoring, diagnostics and ultrasonic imaging.

Sebti Foufou received his Ph.D. degree in computer science in 1997 from the University of Claude Bernard Lyon I, France. He has been working as an associate professor in the Computer Science Department at the University of Burgundy, France. His research interests concern geometric modeling and CAD-CAM topics and surfaces blending, subdivision surfaces, and geometric constraints solving. Currently, he is working as a temporary guest researcher at the National Institute of Standards and Technology, Gaithersburg, Maryland, on smart machining systems, tolerances, assembly modeling, and PLM.

Frédéric Truchetet received his M.S. degree in physics from Dijon University, France, in 1973 and a Ph.D. degree in electronics from the same university in 1977. He was for two years with Thomson-CSF as a research engineer and he is currently full professor in Le2i. His research interests are focused on image processing for artificial vision inspection and particularly on wavelets transform, multiresolution edge detection and image compression. He has authored and co-authored more than 150 international publications, three books, and holds one patent. He is a member of GRETSI, ASTI, IEEE, SPIE, Chairman of SPIE's conference on wavelet applications in industrial processing and member of numerous technical committees of international conferences in the area of computer vision.

Mongi A. Abidi is a professor and deputy director of the Department of Electrical and Computer Engineering, directs activities in the Imaging, Robotics, and Intelligent Systems Laboratory. He received his Ph.D. degree in electrical engineering at The University of Tennessee in 1987, M.S. degree in electrical engineering from the University of Tennessee in 1985, and principal engineer in electrical engineering at the National Engineering School of Tunis, Tunisia in 1981. Dr. Abidi conducts research in the field of 3D imaging, specifically in the areas of scene building, scene description, and data visualization.