Function Representation in Geometric Modeling: Concepts, Implementation and Applications

A.Pasko[1], V.Adzhiev[2], A.Sourin[3], V.Savchenko[1]

[1] Shape Modeling Laboratory

The University of Aizu

Tsuruga, Ikki-Machi, Aizu-Wakamatsu City,

Fukushima, 965 Japan

[2] Department of Computer Science

University of Warwick

Coventry, CV4 7AL, United Kingdom

[3] School of Applied Science

Division of Software Systems

Nanyang Technological University

Nanyang Avenue, Singapore 2263

Geometric modeling using continuous real functions of several variables is discussed. Modeling concepts include sets of objects, operations and relations. An object is a closed point set of *n*-dimensional Euclidean space with a defining inequality $f(x_1, x_2, \ldots, x_n) \geq 0$. Transformations of a defining function are described for the set-theoretic operations, blending, offsetting, bijective mapping, projection, Cartesian product and metamorphosis. Inclusion, point membership and intersection relations are described. In the implemented interactive modeling system, we use high-level geometric language that provides extendibility of the modeling system by input symbolic descriptions of primitives, operations and predicates. This approach supports combinations of representational styles, including constructive geometry, sweeping, soft objects, voxel-based objects, deformable and other animated objects. Application examples of aesthetic design, collisions simulation, NC machining, range data processing, and 3D texture generation are given.

Key words: Geometric modeling - Solid modeling - Real functions - Implicit surfaces - R-functions.

1 Introduction

The use of real functions of several variables for defining geometric objects is quite common in mathematics and computer science. The inequality $f(x_1, x_2, \ldots, x_n) \geq 0$ describes a half space in *n*-dimensional Euclidean space. The equation $f(x_1, x_2, \ldots, x_n) = 0$ specifies an implicit function of *n*-1 variables and describes an orientable (*n*-1)-dimensional surface. The properties of these geometric objects are studied in algebraic and differential geometry. In three-dimensional case, an object defined by the above-mentioned inequality is usually called a solid (or a volume) and an object defined by the equation is called an implicit surface. Functionally represented volumes and surfaces appear to be useful in solid modeling, computer aided geometric design (CAGD), animation, range data processing and volume graphics.

Half spaces defined by algebraic inequalities are used as primitives in constructive solid geometry (CSG) (Requicha 1980, Requicha and Rossignac 1992). Duff (1992) solves such important problems as collision detection and rendering on the basis of interval arithmetic for CSG-trees with primitives bounded by implicit surfaces. The representation of a whole complex object by a single real function also has attracted an interest. An attempt to develop a system of the set-theoretic operations closed on this representation has been made by Ricci (1973). The serious restriction of the proposed method is that functions defining complex objects contain C^1 discontinuity in their domain. The exact analytical definitions of the set-theoretic operations have been proposed in the theory of R-functions (Rvachev 1963, Rvachev 1974) and applied to solving problems of mathematical physics on complicated geometric domains (see Shapiro 1988 and Shapiro 1994a for a survey). Shapiro (1994a) uses R-functions to construct defining functions of regular sets required in CSG. The theory of

R-functions was applied to define several operations on multidimensional geometric objects in (Pasko 1988, Pasko et al. 1993a).

Wang (1984) pointed out the theoretical possibility of deriving the implicit description of the surface swept by the moving solid. Symbolic computations were required to reduce the dimension of variables and to yield the representation in an implicit form.

Although parametric representation is the most common in CAGD, attention is also paid to implicit surfaces because of their closure under some important operations (Hoffmann 1993). Typical operations of this kind are the offsetting and blending (Ricci 1973, Middleditch and Sears 1985, Hoffmann and Hopcroft 1987, Hoffmann 1989, Rockwood 1989, Warren 1989).

Sclaroff and Pentland (1991) have generalized the implicit function representation by introducing deformation defined with a matrix of free vibrations. Such a generalization provides modeling deformations based on physical laws and collision detection for animated objects (Essa et al. 1992). Collision detection for implicit surfaces is also discussed by Baraff (1990), Gascuel (1993) and Snyder et al. (1993).

Several implicit functions have been proposed on the base of field generating skeleton (Blinn 1982, Wyvill et al. 1986a, Bloomenthal and Shoemake 1991). The field defined by superquadrics (Barr 1981) can increase significantly a complexity of the resultant object. These functions are useful in interactive modeling (Bloomenthal and Wyvill 1990) and animation (Wyvill et al. 1986b). Muraki (1991) has applied Blinn's blobby model to approximate the shape of the object defined by sample surface points. Modal deformations and displacement map (Sclaroff and Pentland 1991) as well as polynomial functions (Bajaj et al. 1993) have been also applied for solving the surface reconstruction problem.

Many authors have presented algorithms of polygonization of implicitly defined surfaces or isosurfaces of trivariate functions (Wyvill et al. 1986a, Bloomenthal 1988, Pasko et al. 1988, Schmidt 1993). The similar algorithm was proposed by Lorensen and Cline (1987) for extracting polygonal surfaces from volumetric data. In fact, objects with implicit surfaces and voxel-based objects have the unified models. The only difference is that a tabulated function of three variables is used for the voxel-based objects. Such a view was applied, for example, by Hughes (1992) to implement metamorphosis between two voxelized solids.

Thus, the function representation is widely used in geometric modeling and computer graphics in several different forms. From the other hand, these different models are not closely related to each other and to such well known representations as sweeping, B-rep and CSG (Requicha 1980). This obviously retards further research. The motivation of our work is the need to fill these gaps by constructing as rich as possible system of operations closed on functionally represented objects. Dimension independence of descriptions leads to the possibility of inclusion of 4D and other multidimensional volumes in the set of treated objects. Application of real functions of several variables provides a good base for an interactive modeling system extendable by symbolic descriptions of primitives and operations.

This paper presents a state of the art report of our project, the main objectives of which are:

– Categorization and summary of the geometric concepts required in a functionally based modeling environment;
– Elaboration of a rich system of geometric operations closed on functionally represented objects;
– Treatment of multidimensional and particularly space-time objects in a uniform manner;

– Specification, implementation and application of an interactive geometric modeling system based on the function representation.

In Section 2, we consider geometric concepts as a triple (Objects, Operations, Relations) with their descriptions in terms of real functions of several variables and present new descriptions of some geometric operations. Section 3 deals with interactive geometric modeling based on the function representation. Section 4 is devoted to the application examples. Section 5 summarizes the paper and discusses future work.

2 Geometric concepts and function representation

This section provides a summary of geometric concepts of functionally based modeling environment. We also discuss new descriptions of several geometric operations that we have recently developed.

Let us describe the geometric concepts as a triple:

$$(M, \Phi, W)$$

where M is a set of geometric objects, Φ is a set of geometric operations, and W is a set of relations for the set of objects. Mathematically this triple is an algebraic system. Its application to multidimensional and time-dependent geometric modeling have been studied by Pilyugin et al. (1988), Pasko (1988) and Sourin (1988). Below, we consider the main parts of this triple.

2.1 Objects

We consider geometric objects as closed subsets of n-dimensional Euclidean space E^n with the definition:

$$f(x_1, x_2, \ldots, x_n) \geq 0 \qquad (1)$$

where f is a real continuous function defined on E^n. We call f a *defining function*. The inequality (1) we call a *function representation* (or *F-rep*) of a geometric object. In the three-dimensional case, the boundary of such an object is a so-called "implicit surface". Note that the definition of an object is the inequality (1) with the explicit function of n variables $f = f(x_1, x_2, \ldots, x_n)$ but not the implicit function of n-1 variables $f(x_1, x_2, \ldots, x_n) = 0$. The function can be defined analytically, or with a function evaluation algorithm, or with tabulated values and an appropriate interpolation procedure. The major requirement to the function is to have at least C^0 continuity. There is a classification of points in E^n associated with the closed n-dimensional object with function representation. If $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ is a point in E^n, then:

$f(\mathbf{X}) > 0$ if **X** is inside the object;

$f(\mathbf{X}) = 0$ if **X** is on the boundary of the object;

and $f(\mathbf{X}) < 0$ if **X** is a point outside the object.

Here, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ is a point in E^n. However, after applying operations to such an object, the constructed function may have points with f=0 inside and on the boundary of the object. In some practical applications it can require special treatment. On the other hand, it gives much more freedom for designing operations on objects. Generally speaking, geometric objects defined by the inequality (1) are not reqularized solids required in CSG. The object can have the boundary with dangling portions that are not adjacent to the object's interior. We define objects in multidimensional space for choosing a space of arbitrary dimension in each specific case. For example, if n=4 then (x_1, x_2, x_3) can be space coordinates and x_4 can be interpreted as the time.

Two major types of elements of the set M are basic geometric objects (primitives) and complex geometric objects. A geometric primitive is described by a specific instance of a function chosen from a finite set of possible types. A complex geometric object is a result of operations on primitives. In the modeling system, the finite set of primitives can be defined. However, the possibility of the extension of this set in a symbolic manner should be provided. Actually, this approach allows the modeling system to be initially "empty" and make the user to be responsible for an application oriented filling of the primitive set. We consider this flexibility as one of the major advantages of the F-rep based geometric modeling system.

2.2 Operations

The set of geometric operations Φ includes such operations as:

$$\Phi_i : M^1 + M^2 + \ldots + M^n \to M$$

where n is a number of operands of an operation.

We consider only unary and binary operations in this paper. The result of the operations is also an object from the set M that ensures the closure property of the function representation. Let object G_1 has the definition $f_1(\mathbf{X}) \geq 0$. For unary operations the object G_2 is said to be derived from G_1 as $G_2 = \Phi_i(G_1)$ and is defined by $f_2 = \Psi(f_1(\mathbf{X})) \geq 0$, where Ψ is a continuous real function of one variable. The examples of unary operations are the bijective mapping, affine mapping, projection, offsetting. For binary operations the object G_3 is said to be derived from G_1 and G_2 as $G_3 = \Phi_i(G_1, G_2)$ and is defined by $f_3 = \Psi(f_1(\mathbf{X}), f_2(\mathbf{X})) \geq 0$, where Ψ is a continuous real function of two variables. The examples of binary operations are the set-theoretic operations, blending operations, Cartesian product, metamorphosis. Like with the objects, the user of the F-rep based modeling system is able to introduce any desirable operation by its analytical or procedural description in symbolic form and thus extend the list of operations. In this connection, we shall not attempt to specify the complete set of possible operations on objects, but only introduce the most commonly used types.

The transformations of the function representation associated with operations on an object are described below. We follow here the logic of step by step extension of modeling environment from well-known set-theoretic operations to less familiar metamorphosis.

2.2.1 Set-theoretic operations

The analytical definitions of the set-theoretic operations on functionally described objects have been introduced and studied by Rvachev (1963, 1974) for solving problems of mathematical physics on areas with complex shapes. Rvachev proposed these definitions in order to transform set-theoretic operations on areas with the description (1) in to operations on the defining functions. The resultant object will have the defining function as follows:

$f_3 = f_1 | f_2$ for the union;

$f_3 = f_1 \& f_2$ for the intersection;

$f_3 = f_1 \setminus f_2$ for the subtraction,

where f_1 and f_2 are defining functions of initial objects and **|**,**&** ,\ are signs of so-called R-functions. One of the possible analytical descriptions of R-functions is as follows:

$$f_1 | f_2 = \frac{1}{1+\alpha}\left(f_1 + f_2 + \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2}\right)$$
$$f_1 \& f_2 = \frac{1}{1+\alpha}\left(f_1 + f_2 - \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2}\right) \quad (2)$$

where $\alpha = \alpha(f_1, f_2)$ is an arbitrary continuous function satisfying the following conditions:

$$-1 < \alpha(f_1, f_2) \le 1,\ \alpha(f_1, f_2) = \alpha(f_2, f_1) = \alpha(-f_1, f_2) = \alpha(f_1, -f_2)$$

The expression for the subtraction operation is

$$f_1 \setminus f_2 = f_1 \& (-f_2)$$

Note that with this definition of the subtraction, the resultant object includes its boundary.

If α=1, the functions (2) become

$$f_1 | f_2 = \min(f_1, f_2)$$
$$f_1 \& f_2 = \max(f_1, f_2) \quad (3)$$

This is the particular case described by Ricci (1973). The functions (3) are very convenient for calculations but have C^1 discontinuity when $f_1 = f_2$. If α=0, the functions (2) take the most useful in practice form:

$$f_1 | f_2 = f_1 + f_2 + \sqrt{f_1^2 + f_2^2}$$
$$f_1 \& f_2 = f_1 + f_2 - \sqrt{f_1^2 + f_2^2} \quad (4)$$

The functions (4) have C^1 discontinuity only in points where both arguments are equal to zero. If C^m continuity is to be provided, one may use another set of R-functions:

$$f_1 | f_2 = \left(f_1 + f_2 + \sqrt{f_1^2 + f_2^2} \right) \left(f_1^2 + f_2^2 \right)^{\frac{m}{2}}$$
$$f_1 \& f_2 = \left(f_1 + f_2 - \sqrt{f_1^2 + f_2^2} \right) \left(f_1^2 + f_2^2 \right)^{\frac{m}{2}} \quad (5)$$

Generally, R-functions correspond to standard but not regularized set-theoretic operations. They can result in "interior zeroes" of a defining function. Thus in a general case one can not distinguish between a point of "interior zero" and a boundary point. If regularity of resulting objects is required, the special method of construction of defining functions can be applied (see Shapiro 1994a).

Note that the R-functions can be used for describing geometric primitives. For example, the description of a segment in E^1 can be obtained from the descriptions of two rays as follows:

$$f(x) = (x_1 - b_1) \& (b_2 - x_1)$$

The plot of this function for the expressions (3) is shown in Fig. 1a with α=1, Fig. 1b corresponds to (4) with α=0, and Fig. 1c corresponds to (5) with m=1. It is important to point out that the function in Figs. 1b and 1c does not have points in its domain where the derivative is discontinuous. Such points can cause problems in subsequent operations on objects, especially when blending is used (see Rockwood 1989). Fig.2 illustrates the process of construction of 3D solid with the union and intersection operations.

2.2.2 Blending

Although blending was studied by many researchers, the need in the continuous analytical description of the set-theoretic operations with controllable blend shape existed. Pasko and Savchenko (1994a) have proposed to consider a blending surface as a boundary of an object obtained by the modified set-theoretic operations with the description based on the technique of R-functions. This approach is based on the observation of contours behavior shown in Fig. 3. Note that we use expressions (4) here to define the set-theoretic operation. That is why $f=C$ gives an exact definition of the corner for $C=0$ and smooth contour lines for other C values. The blending set-theoretic operation based on R-functions can be defined as

$$F(f_1, f_2) = R(f_1, f_2) + d(f_1, f_2) \qquad (6)$$

where R is a corresponding R-function, d is a displacement function that has a maximal absolute value $d(0,0)$ and asymptotically approximates a zero value with increasing absolute values of the arguments. We have found that, for example, the following simple form of the displacement function is suitable for blending:

$$d(f_1, f_2) = \frac{a_0}{1 + \left(\frac{f_1}{a_1}\right)^2 + \left(\frac{f_2}{a_2}\right)^2} \qquad (7)$$

It is assumed that the defining functions for both objects have the distance property. The proposed displacement function is not the only one possible. Other Gaussian-like functions can be designed for specific applications. Applying equations (4), (6) and (7) we can get, for example, the final description form of the blending intersection operation:

$$F(f_1, f_2) = f_1 + f_2 + \sqrt{f_1^2 + f_2^2} + \frac{a_0}{1 + \left(\frac{f_1}{a_1}\right)^2 + \left(\frac{f_2}{a_2}\right)^2} \qquad (8)$$

This definition provides highly intuitive shape control of added material, subtracted material, and variable radius blends. It was used to generate aesthetic blends defined by hand-drawn strokes (see Section 4). Constant-radius blending is connected with offsetting operation and is discussed below. Fig.4a illustrates an application of the blending set-theoretic operations. The basic block is not defined as a single primitive but as an intersection of six halfspaces to control the shape of edges in further blending. Different values of the displacement function parameters were assigned to the different edges. The prominent front edge of the resultant object was defined by added material blending. Other edges were defined by subtracted material blending. Multiple blending operations were applied in the corners. The small cylindrical hole was made after blending to show that no problem occurs when applying the set-theoretic operations to the blended object. This is ensured by the continuity of the function (8).

2.2.3 Offsetting

To generate expanded or contracted versions of an initial object one can apply the positive and negative offsetting operation respectively. The descriptions of the following three offsetting operations have been proposed in (Pasko and Savchenko 1994b):

1. Iso-valued offsetting with $F=f(\mathbf{X})+C$, where negative constant C defines the negative offset, and positive C defines the positive offset. Fig.4b shows an application of this operation to a 3D solid.
2. Offsetting along the normal with $F=f(\mathbf{X}+D\mathbf{N})$ for the positive and $F=f(\mathbf{X}-D\mathbf{N})$ for the negative offsetting, where D is the given distance value, and **N** is a gradient vector of the function f in the point with **X** coordinates.
3. Constant-radius offsetting with $F=\max(f(\mathbf{X}'))$ for the positive and $F=\min(f(\mathbf{X}'))$ for the negative offsetting, where **X'** is vector of coordinates of points belonging to the sphere of the given radius D and the center at **X**. Note that continuity and distance property of the defining function of the object are also used here. The procedure of the

constant-radius blending of the object's convex edges includes consequently applied negative and positive constant-radius offsetting (Rossignac and Requicha 1984). This procedure applied to 2D solid is illustrated in Fig.4c.

2.2.4 Bijective mapping

Let Φ_i be defined by coordinate transformations

$x_j' = \varphi_j(x_1, x_2, \ldots, x_n),\ j = 1, \ldots, n$

where φ_j are continuous real functions. We assume also that inverse functions φ_j^{-1} exist. Then the resultant object is described as

$$G_2 : f_1(\varphi_1^{-1}(x_1', x_2', \ldots, x_n'), \ldots, \varphi_{n-1}^{-1}(x_1', x_2', \ldots, x_n')) \geq 0 \qquad (9)$$

The examples of such a bijective mapping are the tapering, twisting, bending (Barr 1984) and modal deformations (Sclaroff and Pentland 1991). The twisting of a solid constructed by the set-theoretic operations is shown in Fig.5a. Another example is the mapping of an object, defined in an arbitrary coordinate system, to a Cartesian coordinate system. The inverse functions φ_j^{-1} from (9) for the mapping of an object from the cylindrical coordinate system to the Cartesian one will be defined as:

$$\begin{aligned} x_1 &= \varphi_1^{-1}(x_1', x_2', x_3') = \sqrt{x_1'^2 + x_2'^2} \\ x_2 &= \varphi_2^{-1}(x_1', x_2', x_3') = \arctan(x_2'/x_1') \\ x_3 &= \varphi_3^{-1}(x_1', x_2', x_3') = x_3' \end{aligned} \qquad (10)$$

where

$x_1 = \rho, x_2 = \theta, x_3 = z$ are cylindrical coordinates

$x_1' = x, x_2' = y, x_3' = z$ are Cartesian coordinates

This mapping can be applied to describe an object defined by rotational sweeping (see the Cartesian product operation described below).

2.2.5 Affine mapping

The affine mapping is an important specific case of the bijective mapping. Let Φ_i be the affine mapping defined by the equality $\mathbf{X}' = \mathbf{AX} + \mathbf{C}$, where $\mathbf{X} = (x_1, x_2, \ldots, x_n)^T, \mathbf{X}' = (x_1', x_2', \ldots, x_n')^T, \mathbf{C} = (c_1, c_2, \ldots, c_n)^T$,
and $A = \{a_{ij}\}$ is the matrix of transformation with dimensionality $n \times n$ and $\det \mathbf{A} \neq 0$. Then the definition (9) is changed to:

$$G_2 : f_1(\mathbf{A}^{-1}(\mathbf{X}' - \mathbf{C})) \tag{11}$$

In general case, elements of the matrix **A** and the vector **C** can be functions of coordinates **X**. For example, if they are time dependent, then some time-dependent affine mapping is defined. It can be the movement along a line, rotation and scaling in time, delay and bringing forward, complex movement with time mapping, etc. The unification of mappings of space coordinates and the time allows an effective description of complex geometric processes unifying models used in a description of geometric volumes and complex movements in animation.

Next we consider the operation of projection where det**A**=0.

2.2.6 Projection

R-functions can also be applied for approximated describing the projection operation from E^n to E^{n-1} that does not have an inverse transformation Φ^{-1}. Let us define

$$G_1 \subset E^n \quad : f_1(x_1, x_2, \ldots, x_i, \ldots, x_n) \geq 0$$
$$G_2 \subset E^{n-1} : f_2(x_1, x_2, \ldots, x_{i-1}, x_{i+1} \ldots, x_n) \geq 0$$

and G_2 is a projection of G_1 to E^{n-1}. The object G_2 can be defined as a union of sections of G_1 by hyperplanes $x_j = C_j$ where $C_{j+1} = C_j + \Delta x_i, \quad j = 1, N$ and $C_1 = X_{i\min}$.

Let $f_{1j} = f_1(x_1, x_2, \ldots, x_{i-1}, C_j, x_{i+1}, \ldots, x_n)$ be a defining function for a section. Then the defining function for the projection with $\Delta x_i \to 0$ can be expressed as:

$$f_2 = f_{11} \mid f_{12} \mid \ldots \mid f_{1j} \mid \ldots \mid f_{1N} \tag{12}$$

Numerical procedures for this function evaluation are discussed elsewhere (Pasko 1988). Fig.5b illustrates a projection from E^3 to E^2. One of the applications of this operation is a description of a volume swept by a moving solid as a projection of 4D object to E^3.

2.2.7 Cartesian product

Let $G_1 \subset E^k$ and $G_2 \subset E^m$.

We define G_3 as a Cartesian product of G_1 and G_2:

$$G_3 = G_1 \times G_2 = \{(x_1, x_2, \ldots, x_n) | (x_1, x_2, \ldots, x_k) \in G_1, (x_{k+1}, x_{k+2}, \ldots, x_n) \in G_2\}$$

where $G_3 \subset E^n$ and n=k+m. The defining function for G_3 can be obtained using R-functions:

$$f_3(x_1, x_2, \ldots, x_n) = f_1(x_1, x_2, \ldots, x_k) \& f_2(x_{k+1}, x_{k+2}, \ldots, x_n) \qquad (13)$$

Swept objects can be defined with help of the Cartesian product and the bijective mapping. A 3D object defined by the rotational sweeping is shown in Fig.5c. Firstly, the 3D object was constructed as Cartesian product of 2D solid and 1D segment. Then bijective mapping (10) was applied to this 3D object. In other words, its x-coordinate was interpreted as the angle and y-coordinate as the radius of a cylindrical coordinate system. Another application is a definition of time dependent geometric objects on some time interval by the Cartesian product of a static object and a time segment or a ray. For example,

$G_1 \subset E^3 : f_1(x, y, z) \geq 0$ is a static object in E^3

$G_2 \subset E^1 : (t - t_1) \& (t_2 - t) \geq 0$ is a time segment $[t_1, t_2]$ (14)

$G_3 \subset E^4, G_3 = G_1 \times G_2 : f_1(x, y, z) \& ((t - t_1) \& (t_2 - t)) \geq 0$

G_3 is a time dependent geometric object being activated at the time t_1 and terminated at the time t_2 in space E^4 with coordinates (x,y,z,t).

2.2.8 Metamorphosis

We consider the metamorphosis as a binary operation on two objects G_1 and G_2 defined in E^{n-1}. The resultant object G_3 is defined in E^n and described as

$$f_3(x_1, x_2, \ldots, x_n) = f_1(x_1, x_2, \ldots, x_{n-1}) \cdot (1 - g(x_n)) + f_2(x_1, x_2, \ldots, x_{n-1}) \cdot g(x_n) \quad (15)$$

where $g(x_n)$ is a positive continuous function, $g(x_n{}^0) = 0$, and $g(x_n{}^1) = 1$. It means that G_1 is a section of G_3 by the hyperplane $x_n = x_n{}^0$ and G_2 is a section of G_3 by the hyperplane $x_n = x_n{}^0$ in E^n. For n=4, an object G_3 can be thought as a time dependent object reconstructed from its two instances at different time moments. Fig.5d shows several time steps of metamorphosis of a 3D solid to another one. For n=3, this operation generates a 3D solid reconstructed from its two planar cross-sections that can be useful in tomography and range data processing.

2.3 Relations

We will consider only binary relations as the subsets of the set M^2. The examples of binary relations are the inclusion, point membership, interference or collision. Similarly, like with the objects and operations, we give the user a possibility to extend the set of relations by symbolic definition of their predicates.

2.3.1 Inclusion relation

This relation is described as $G_2 \subset G_1$ and means that the object G_2 is included in G_1. If G_2 is the point P the relation can be described by the following bivalued predicate:

$$S_2(P, G_1) = \begin{cases} 0, \text{ if } f_1(\mathbf{X}) < 0 & \text{for } P \notin G_1 \\ 1, \text{ if } f_1(\mathbf{X}) \geq 0 & \text{for } P \in G_1 \end{cases} \quad (16)$$

2.3.2 Point membership relation

Let iG_1 be the interior of G_1 and bG_1 be the boundary of G_1. The point membership relation is described by the 3-valued predicate:

$$S_3(P, G_1) = \begin{cases} 0, \text{ if } f_1(\mathbf{X}) < 0 & \text{for } P \notin G_1 \\ 1, \text{ if } f_1(\mathbf{X}) = 0 & \text{for } P \in bG_1 \\ 2, \text{ if } f_1(\mathbf{X}) > 0 & \text{for } P \in iG_1 \end{cases} \qquad (17)$$

This predicate can be correctly evaluated for G_1 with no interior zeroes. Note that the set-theoretic operations with R-functions correspond to the operations of 3-valued logic over predicates S_3 but not to the Boolean logic over predicates S_2.

2.3.3 Intersection relation

The relation is defined by the bivalued predicate:

$$S_c(G_1, G_2) = \begin{cases} 0, \text{ if } G_1 \cap G_2 = \varnothing \\ 1, \text{ if } G_1 \cap G_2 \neq \varnothing \end{cases} \qquad (18)$$

A function $f_3(\mathbf{X}) = f_1(\mathbf{X}) \& f_2(\mathbf{X})$ defining the result of the intersection can be used to evaluate S_c. It can be stated that $S_c = 0$ if $f_3(\mathbf{X}) < 0$ for any point of E^n (Rvachev 1967). This definition leads to the collision detection algorithm discussed in Section 4.

3 Interactive geometric modeling based on the function representation

3.1 A machine representation and a user representation

In the previous section we introduced F-rep as a mathematical notion of an abstract nature allowing to define basic geometric concepts - objects, operations, and relations - in terms of real functions of several variables. This section is devoted to describing our experience of building an interactive geometric modeling system based on the F-rep notion. Accordingly, the very notion "representation" will be treated in more applied sense as concerned with the computer's and user's manipulation and interpretation of it in our current modeling system.

We share Snyder's belief that "the representation to be the part of a geometric modeling system which most determines its quality" (Snyder 1992) and consider F-rep as an essentially "user representation" allowing to specify user's geometric model in the symbolic form. We explicitly distinguish it from a lower-level "machine representation" which can be present inside the system. This lower level can be in the form of a generalized CSG-type tree with, in turn, other levels, such as collections of polygons. The corresponding set of procedural tools forms a kernel geometric modeler. Pasko et al. (1993a) give the VDM-specification of such a modeler which formally defines principal data structures of a "machine representation" as well as operations over them. This modeler together with a visualization subsystem serve as a basis for building an interactive geometric modeling system.

3.2 Geometric language and the example of modeling

F-rep as a user representation serves as a base for a high-level geometric language which is the user's instrument for modeling specification. To realize how significant features of F-rep are reflected in the geometric language, let us consider the corresponding modeling program (Fig.6). This program creates a model of cycled metamorphosis between the following four geometric objects which are normally modeled through different representational styles:

1. Constructive object defined with help of the set-theoretic operations on primitives;
2. Swept object defined by the Cartesian product and the subsequent bijective mapping;
3. Voxel-based object built by manual sculpting similar to (Galyean and Hughes 1991) with the subsequent trilinear interpolation providing C^0 continuity to make this volumetric tea-pot to be a "legal" F-rep object;

4. "Blobby" object as a representative of objects with analytically defined implicit surfaces.

Finally, metamorphosis itself is modeled with using the objects described as "key volumes" and eventual getting necessary intermediate volumes in accordance with (15). Fig.7 represents the frames of the computer film corresponding to the specification of the resultant 4D geometric object *gob_metamorp* from Fig.6.

The program being interactively introduced during a modeling session consists of the following parts:

– Geometric model

– Geometric types

– Environment

Each part can be defined and changed irrespective of other ones in the process of interactive modeling work. The brief description of these parts is given below.

3.2.1 Geometric Model

This is actually a parametrized specification of geometric objects themselves which is intimately related with F-rep. Geometric objects are given by their defining functions. In keeping with the mathematical framework described in Section 2, each function defining complex geometric object *gob_<name>* is built as a rather traditional mathematical expression with using symbols of coordinate variables x_i, geometric primitives *pob_<name>*, numerical constants and parameters, arithmetic operations, standard algebraic functions (sin, cos, log, min, max, etc.), and built-in set-theoretic operations implemented by R-functions (| - for union, & - for intersection, \ - for subtraction, and @ - for Cartesian product). One can make choice of the type of R-functions by setting *r_alpha* parameter. If built-in *pob_block_3D* was defined with help of "min-max" system of R-functions with *r_alpha*=1 (3), the user can introduce new one defined with help of the system of R-functions (4) with better continuity properties. Such operators of structured programming as "*if-then-else*" and "*while-do*" together with complete set of logical functions provide a proper programming flexibility allowing to define complex and non-traditional geometric transformations and relations.

Note, besides built-in conventional primitives and operations one can introduce new ones during a modeling session. For instance, if there is a need in a "block" with blended edges, one can define the following "blending intersection" transformation based on (8):

tr_bl_int((*gob*|*pob*) *g*1, (*gob*|*pob*) *g*2, (real)*a*1, (real)*a*2, (real)*a*3) =

$g1+g2\text{-sqrt}(g1^2+g2^2)+a1/(1+(g1/a2)^2+(g2/a3)^2)$;

Then this operation can be applied to half-spaces to get a block with smooth edges and corners.

3.2.2 Geometric Types

Each coordinate variable x_i can be associated with a certain "geometric type". These types establish conventions governing their semantics by giving geometric interpretation of x_i. This interpretation can be important during exploring the geometric model, in particular in visualization. There are the following geometric types in the current version of our system:

– "*constant*" : Variables of this type are assigned numerical values to define a single cross-section, such as $x_i = const$;

– "*g*" : These variables define a group of constants corresponding to several cross-sections;

– "*x*", "*y*", "z" : These are coordinate variables corresponding to axes in the 3D Cartesian coordinate system;

– "*t*" : Variables of this type model a course of time with a possibility of its incremental or decremental changing that can be used in animation. Time-dependent geometric objects are called geometric processes in our system;

– "*v*", "*w*" : These correspond to the additional V-axis and W-axis for building a geometric spreadsheet with elementary images in cells to support so-called inductive approach to multivariate function visualization (Pasko et al. 1992);

– "*c*" : Variables of this type are used for mapping to colors within a spectral range.

Note, that geometric types "*constant*" and "*g*" can also be assigned to the defining function of the resultant geometric object. Normally, the zero value sets boundaries of a geometric object. If "*g*" type is assigned, visualization of corresponding isolines and isosurfaces can be very useful for exploring the features of the defining function (see Fig.3 as an example).

3.2.3 Environment

This section is intended for concretizing those parameters being present in the "Geometric Model" which were denoted by their abstract names. The first sub-section defines ranges $[x_{i\,\min}, x_{i\,\max}]$ of coordinate variables to set boundaries of a modeling space. Then an incremental interval x_i_*delta* for coordinate variable with geometric type "*t*" must be defined. In the example, the value x_i_*delta*=0.5 lets us get one intermediate frame between each pair of "key volumes". The next sub-section deals with assigning values of numerical parameters including arrays. The user can specify necessary visualization parameters that essentially depend on the assignment of geometric types to coordinate variables.

3.3. Advantages of Function Representation as a Base for Interactive Modeling System

We use (with some modifications) the following three criteria for evaluation of F-rep that were introduced by Snyder (1992) for a user representation as a special adaptation of the categories from [Requicha 1980].

1) Ease of specification.

This is the first and the most crucial criterion for evaluation of the quality of a user representation which basically assesses how efficiently the users can define and change their geometric model.

The function representation is closed, meaning that further operations can be applied to results of previous ones. This representation is also uniform in the sense that it supports combinations of representational styles to describe objects of traditionally different nature. These two properties were substantiated above.

It results in the conclusion about the higher abstraction level as regards many other representations. Uniform definition for objects in spaces of various dimensions as well as uniform representation of static and time-dependent objects are especially attractive. Moreover, multidimensionality can be interpreted in the interesting and natural for this representation way through the concept of geometric types.

It is obvious that F-rep, because of its analytical nature, provides such important for user's modeling work categories as compactness and accuracy of a model.

Just because of thousand-year tradition of analytical description of geometry, we consider F-rep as natural for users we are oriented towards. Even elementary knowledge of analytical geometry and brief training let the users connect the way they think about geometric shapes with symbolic descriptions in the form of analytical expressions. The very nature of F-rep lets the users easily change some parameters in the analytical model and observe the visual or computational effect setting the understandable correspondence between analytical description and model's behavior. So, editability and controllability of description are naturally provided.

At last but not at least one should mention extendibility meaning the possibility of introducing new primitives and operations not only by procedural defining but also by symbolic one during a modeling session. It lets creating a specific modeling system for particular application areas and even for particular users who are able to change the system in some item if they want. This is a base for "Empty Case" technology of geometric modeling (Pasko et al. 1993b) which supposes working process of "absolutely first user" who has to create his personal geometric modeling system defining necessary primitives and transformations in a symbolic manner. This seems to be especially valuable in educational perspective, and corresponding project is being realized at the University of Aizu.

2) Renderability.

We consider F-rep as a quite suitable for providing fast visual feedback given to the user. It is achieved through conversion to a polygonal mesh. The original algorithm of polygonization of an isosurface $f(x_1, x_2, x_3) = 0$ (Pasko et al. 1988) was used to generate the frames in Fig.7. This algorithm can be easily decomposed in order to map on parallel computer architecture. Higher quality rendering is also possible with ray casting which also undergoes parallelization very well. Compactness of F-rep allows to run rendering software even on parallel computers with not very large size of distributed memory (e.g. transputer networks).

3) Analyzability.

Geometric queries such as point inclusion are simple for the function representation. It helps to compute physical quantities about the shape (volume, moments of inertia) with well-known algorithms. Collision detection can be realized by a maximum search procedure (see Section 2.3

and the example in Section 4). On the other hand, finding curves of intersection of surfaces require slow numerical algorithms.

3.4 Perspective interactive environment

The function representation as a high-level user representation fits very well for "exploratory geometric modeling". It means not simply describing geometric model whose properties and features are preliminary given, but rather introducing new geometric objects and transformations with subsequent exploring their characteristics and behaviour in an interactive manner. This creative process is similar to a traditional scientific investigation of a physical phenomenon when experimenting with the model created and observing its behaviour under changeable conditions are performed.

We think the corresponding interactive environment can be built on the basis of the "definitive-based" programming paradigm (Beynon 1989) and the agent-oriented framework which provides easy interactive way for modifying the specification (both parameters in defining functions and functions themselves) with indivisible propagation of changing any dependent entities.

The corresponding implementation work is in progess. Adzhiev et al. (1994) propose LSD-specification of a geometric modeling system. This specification describes within agent-oriented modeling framework both the interactions of the user with the system and the interactions between the principal components of the system itself. This specification can serve as a basis for parallel implementation of the interactive modeling system with an advanced graphical user interface.

4 Application examples

Here we give several examples of application of geometric modeling software based on the function representation principles.

4.1 Aesthetic design

Although the blending operations described in 2.2 provide easy shape control, they seem too indirect for the aesthetic blending. Rather than to change numeric parameters, a designer prefers to define the shape of an aesthetic blend with a single hand-drawn stroke. These points are then used to estimate values of parameters which define the blend as close to this stroke as possible. This is illustrated in Fig.8. The stroke belongs to a certain plane. Points of hand-drawn strokes are assumed to belong to the blending surface. These constraints determine a system of nonlinear equations:

$$F(f_1(x_i, y_i, z_i), f_2(x_i, y_i, z_i), a_0, a_1, a_2) = 0, \quad i = 1, N$$

where f_1 and f_2 are defining functions of initial objects, F defines the blending set-theoretic operation (6), and N is a number of points. To find out the best estimation of a_0, a_1 and a_2 in the sense of least squares, we applied a simple quadratic search using random points for the initial estimate.

4.2 Simulation of collisions

Several application problems (e.g., air and water quality control, collision dynamics of bodies in celestial mechanics, computer games) deal with irregularly shaped interacting solids. Fig.9 illustrates the simulation of colliding particles sticking to each others. It presents collisions of the noisy block, small spheres, torus and noisy ellipsoid. The spheres having collision point with the noisy block are stuck to it by the blending union operation. The surfaces of the block and the ellipsoid were generated using solid noise that is discussed below. All bodies have changeable orientation in the space and interact each other as rigid bodies. The collision detection algorithm is based on the intersection relation defined in 2.3. To find out a collision point of two particles we apply maximum searching algorithm to the defining function of particles intersection. The admissible domain is detected using bounding spheres. Then we use spiral quadratic search within this domain to detect a point with positive or zero function value. The simulation algorithm was implemented on transputers T805 in OCCAM-2 (INMOS 1988) and is described in Savchenko and Pasko (1993). It has simulated 1320 time steps for 35 small spheres. Steps 1020 and 1023 in Fig.9 illustrate the collision event between the torus and the newly formed object.

4.3 Modeling NC machining

The set-theoretic operations between moving solids can be used to model a process of NC machining. The result can be defined with the set-theoretic subtraction between the workpiece model and the swept model of the moving tool. Sourin and Pasko (1994) have proposed the procedural function representation of a swept solid with an envelope surface for a parametrically defined trajectory. Fig.10 illustrates its application to modeling the time-dependent set-theoretic operations.

4.4 Reconstruction of solids from cross-sections

Tomography, range data processing, and other applications need reconstruction of a solid from its given cross-sections. The metamorphosis operation (15) describes a solid using defining functions of two parallel cross-sections. Generation of a defining spline function of a cross-section given by its contour points is proposed by Savchenko et al. (1994). Reconstruction process is illustrated by Fig.11. Note that this approach is capable of generating highly concave and branching solids automatically.

4.5 Three-dimensional texture generation

To obtain 3D textures on constructive solids, Pasko and Savchenko (1993) have proposed to apply the set-theoretic and other operations to a solid defined by a "solid noise" function. Fig.12a shows an irregularly shaped vase. The initial vase was designed using aesthetic blending (see above) of several ellipsoids. Irregular shape was obtained by metamorphosis between the initial vase and solid noise. Fig.12b shows fur obtained by offsetting and set-theoretic intersection applied to an initial solid and fur strands defined using solid noise.

5 Summary and future work

In conclusion, we would like to summarize the main advantages of F-rep and to discuss future research developments.

The function representation for geometric objects offers a number of advantages:

– Higher abstraction level as regards other known representations is provided. Combinations of representational styles, including constructive geometry, sweeping, soft (blobby) objects, and voxel-based objects, are supported.

– Possibility of the symbolic definition of new primitives, operations and predicates is naturally provided. A symbolic description of a complex geometric object as a result of modeling can be generated too.

– Uniform representation for objects defined in spaces of various dimensions. Dimension increasing (Cartesian product) and dimension decreasing (projection) operations are supported. Static objects and time-dependent geometric processes are described uniformly with the time concerned as one of coordinates.

– Convenience of designing application algorithms especially for parallel computers. Compactness of the representation allows to implement application algorithms even on parallel computers with not very large size of distributed memory, for example, on a transputer network.

There are several problems of using F-rep. Evaluation of defining function in a given point is a time consuming task. Moreover, if R-functions with square roots are applied, calculations become slower. The halftone images presented in the paper have been produced on Silicon Graphics Indigo2 using ray-casting. Average time for 200x200 image calculation in double precision is 20-90 sec. One can suppose that numerical stability of nested square roots (see Eqs.2) is questionable. Our numerical experiment has shown that R-union (4) applied to 10000 different spheres (calculated in double precision) gives error 0.2*10sup(-15) for zero value of the defining function in a boundary point. Although graphic workstations provide acceptable response time even for ray-casting, we see the final solution in parallel computing and special hardware. Parallel implementation of the polygonization algorithm (Savchenko and Pasko 1994) improved performance of computations and practically linearly scaled with the number of processors.

In practical systems, conversion from boundary representations may be required. Shapiro (1994b) has proposed that the way to convert B-rep to F-rep is first to convert it to a constructive representation using standard (non-regularized) set operations, and then to F-rep using R-functions. The problem of converting B-rep to the standard set operations is similar to B-rep/CSG conversion (Shapiro and Vossler 1993) but not the same. This problem needs further investigation.

Now, there is not direct connection between F-rep and parametric representations. Because parametric surfaces are very suitable for interactive geometric design, we try to incorporate these models in the function representation. Spline controlled deformations of constructive solids are also very attractive.

The function representation requires users to define a model in a highly abstract way. Coordinate variables, numerical constants and parameters, arithmetic operations and standard algebraic functions are always needed to be explicitly defined by the users. It can be difficult for general users to define objects in this way.

Although symbolic description is a powerful modeling tool, adequate graphical user interface has to be specified for F-rep based modeling. However, the possibility to extend a modeling system with symbolic descriptions of new elements has to be preserved. Interrelations with computer algebra systems will be under research. Now we are applying the function representation to virtual reality applications.

Acknowledgements

Our sincere thanks to Dr. V.V.Pilyugin from MEPI (Russia) and Dr. W.M.Beynon from University of Warwick (UK) for fruitful theoretical discussions. We wish to thank Dr. V.A.Galatenko and V.K.Nikolaev from SRISA RAS (Russia) for software support. The authors also wish to acknowledge referees for their helpful comments.

References

Adzhiev VD, Beynon WM, Pasko AA (1994) Interactive geometric modeling based on R-functions: an agent-oriented approach, Set-theoretic Solid Modeling: Techniques and Applications, CSG'94 Conference Proceedings, Information Geometers, Winchester, UK, 253-272.

Bajaj C, Ihm I, Warren J (1993) Higher-order interpolation and least-squares approximation using implicit algebraic surfaces, ACM Transactions on Graphics, 12(4):327-347.

Barr A (1981) Superquadrics and angle-preserving transforms, IEEE Computer Graphics and Applications, 1(1):11-23.

Barr AH (1984) Global and local deformations of solid primitives, Computer Graphics, 18(3):21-30.

Baraff D (1990) Curved surfaces and coherence for non-penetrating rigid body simulation, Computer Graphics, 24(4):19-28.

Beynon WM (1989) Evaluating definitive principles for interaction in graphics, New Advances in Computer Graphics, Proceedings of CG International'89, 291-303.

Blinn JF (1982) A generalization of algebraic surface drawing, ACM Transactions on Graphics, 1(3):235-256.

Bloomenthal J (1988) Polygonization of implicit surfaces, Computer Aided Geometric Design, 5(4):341-355.

Bloomenthal J, Wyvill B (1990) Interactive techniques for implicit modeling, Computer Graphics, 24(2):109-116.

Bloomenthal J, Shoemake K (1991) Convolution surfaces, Computer Graphics, 25(4):251-256.

Duff T (1992) Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry, Computer Graphics, 26(2):131-138.

Essa I.A., Sclaroff S., Pentland A. (1992) A unified approach for physical and geometric modeling for graphics and animation, Computer Graphics Forum, 11(3):129-138.

Galyean TA, Hughes JF (1991) Sculpting: an interactive volumetric modeling technique, Computer Graphics, 25(4):267-274.

Gascuel MP (1993) An implicit formulation for precise contact modeling between flexible solids, Computer Graphics Proceedings, Annual Conference Series, 313-320.

Hoffmann C, Hopcroft J (1987) The potential method for blending surfaces and corners, Geometric Modeling, G.Farin, Ed., SIAM, Philadelphia, 347-365.

Hoffmann CM (1989) Geometric and solid modeling, Morgan Kaufmann, San Mateo, California.

Hoffmann CM (1993) Implicit curves and surfaces in CAGD, IEEE Computer Graphics and Applications, 13(1):79-88.

Hughes JF (1992) Scheduled Fourier volume morphing, Computer Graphics, 26(2):43-46.

INMOS (1988) OCCAM-2. Reference manual, INMOS Limited, UK.

Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3D surface construction algorithm, Computer Graphics, 21(4):163-169.

Middletich AE, Sears KH (1985) Blend surfaces for set theoretic volume modeling systems, Computer Graphics, 19(3):161-170.

Muraki S (1991) Volumetric shape description of range data using Blobby model, Computer Graphics, 25(4):227-235.

Pasko A (1988) Conceptual and instrumental tools for a direct method of multidimensional geometric problems solving with a computer, Ph.D. dissertation, Moscow Engineering Physics Institute, Russia.

Pasko A, Pilyugin V, Pokrovskiy V (1988) Geometric modeling in the analysis of trivariate functions, Computers & Graphics, 12(3/4):457-465.

Pasko A, Adzhiev V, Prostakov I (1992) Multivariate functions visualization: the inductive approach, Proceedings of Third Eurographics Workshop on Visualization in Scientific Computing, Viareggio (Italy) 27-29 April 1992, 303-316.

Pasko A, Savchenko V, Adzhiev V, Sourin A (1993a) Multidimensional geometric modeling and visualization based on the function representation of objects, TR 93-1-008, The University of Aizu, Japan.

Pasko AA, Savchenko VV, Adzhiev VD (1993b) Educational project: Empty Case technology of geometric modeling, EDUGRAPHICS'93 and COMPUGRAPHICS'93, Combined Proceedings, Ed. H.P.Santo, ACM, Portugal, 6-11.

Pasko AA, Savchenko VV (1993) Solid noise in the constructive solid geometry, EDUGRAPHICS'93 and COMPUGRAPHICS'93, Combined Proceedings, Ed. H.P.Santo, ACM, Portugal, 351-357.

Pasko AA, Savchenko VV (1994a) Blending operations for the functionally based constructive geometry, Set Theoretic Solid Modeling: Techniques and Applications, CSG 94 Conference Proceedings, Information Geometers, Winchester, UK, 151-161.

Pasko AA, Savchenko VV (1994b) Offsetting operations for the functionally based constructive geometry, 6th International Conference on Engineering Computer Graphics and Descriptive Geometry, Tokyo, Japan (to appear)

Pilyugin VV, Adzhiev VD, Pasko AA, Sourin AI (1988) Theoretical issues of dynamic and multidimensional geometric modeling with a computer, Problems of informational systems, No.5, 1988, ICSTI, Moscow, 3-35.

Requicha AAG (1980) Representations for rigid solids: theory, methods, and systems, Computing Surveys, 12(4):437-464.

Requicha AAG, Rossignac JR (1992) Solid modeling and beyond, IEEE Computer Graphics and Applications, 12(9):31-44.

Ricci A. (1973) A constructive geometry for computer graphics, The Computer Journal, 16(2):157-160.

Rockwood AP (1989) The displacement method for implicit blending surfaces in solid models, ACM Transactions on Graphics, 8(4):279-297.

Rvachev VL (1963) On the analytical description of some geometric objects, Reports of Ukrainian Academy of Sciences, 153(4):765-767.

Rvachev VL (1974) Methods of logic algebra in mathematical physics, Naukova Dumka, Kiev.

Savchenko VV, Pasko AA (1993) Simulation of dynamic interaction between rigid bodies with time-dependent implicitly defined surfaces, Parallel Computing and Transputers, ed. D.Arnold, IOS Press,122-129.

Savchenko VV, Pasko AA, Okunev OG, Kunii TL (1994) Function representation of solids reconstructed from scattered surface points and contours, TR 94-1-032, The University of Aizu, Japan.

Savchenko VV, Pasko AA (1994) Parallel polygonization of implicit surfaces on transputers, Transputer Research and Applications 7, Ed. H.Arabnia, IOS Press, pp.22-30.

Schmidt MFW (1993) Cutting cubes - visualizing implicit surfaces by adaptive polygonization, Visual Computer, 10(2):101-115.

Sclaroff S, Pentland A (1991) Generalized implicit functions for computer graphics, Computer Graphics, 25(4):247-250.

Shapiro V (1988) Theory of R-functions and applications: a primer, TR CPA88-3, Cornell University.

Shapiro V, Vossler DL (1993) Separation for boundary to CSG conversion, ACM Transactions on Graphics, 12(1):35-55.

Shapiro V (1994a) Real functions for representation of rigid solids, Computer Aided Geometric Design, 11(2):153-175.

Shapiro V (1994b) On B-rep to F-rep conversion, private communication

Snyder JM (1992) Generative modeling for computer graphics and CAD, Academic Press.

Snyder JM, Woodbury AR, Fleischer K, Currin B, Barr AH (1993) Interval methods for multi-point collisions between time-dependent curved surfaces, Computer Graphics Proceedings, Annual Conference Series, 321-334.

Sourin AI (1988) Mathematics and software for problems of time-dependent geometry, Ph.D. dissertation , Moscow Engineering Physics Institute, Russia.

Sourin AI, Pasko AA (1994) Geometric modeling for NC machining with functionally represented moving solids, IEEE TENCON'94 Conference, Singapore (to appear)

Wang WP (1984) Solid geometric modeling for mold design and manufacture, TR 44, Cornell University, Ithaca, New York.

Warren J (1989) Blending algebraic surfaces, ACM Transactions on Graphics, 8(4):263-278.

Wyvill G, McPheeters C, Wyvill B (1986a) Data structure for soft objects, The Visual Computer, 2(4):227-234.

Wyvill B, McPheeters C, Wyvill G (1986b) Animating soft objects, Visual Computer, 2(4):235-242.

{ Metamorphosis }
{ "Geometric model" section }

```
{ Setting the system of R-functions ("|", "&", "\", "@") }
r_alpha = 1.
{ Primitive "block" }
pob_block_3D(a0,b0,c0,a,b,c,)=   (x1-a0 & a0+a-x1)& (x2-b0 & b0+b-x2)& (x3-c0 &c0+c-x3);

{ CSG object }
{ Basic block }
gob_bl1 = p_block_3D(-9.,-8.,-9.,18.,16.,18.);

{ Vertical cylinder }
gob_cyl1 = (r1**2-x1**2-x3**2) & 10.5-x2) & x2+10.5;

{ Horizontal infinite cylinder }
gob_cyl2 = r2**2-x1**2-x2**2;

{ Smaller block with infinite x1-dimensionality }
gob_bl2 = (x2+d1 & d1-x2) & (x3+d1 & d1-x3);

{ final CSG object }
gob_csg = ( ( gob_bl1 | gob_cyl1 ) \ gob_cyl2 ) \ gob_bl2;

{ Swept object built with help of cartesian product and bijective mapping }
gob_swept = ( r3**2 - (x1-c1*SIN(-1+x2/c2)**2 - (x3-c1*COS(-1+x2/c2)**2 ) @ ( x2+10. & 10.-x2);

{ Voxel-based object defined by 3D array in a file with subsequent interpolation }
gob_vox = INTERPOLATE("trilinear", "teapot.dat", x1,x2,x3);

{ blobby object }
gob_blobby = b;
WHILE ( i <= 8 ) DO
gob_blobby = gob_blobby + a[i]*EXP(-SQRT( (x1-px[i])**2+ (x2-py[i])**2+ (x3-pz[i])**2);

{Metamorphosis between all objects }
gob_metamorp =      IF ( 0 <= x4 < = 1 )
                    THEN gob_csg*(1-x4) + gob_swept*x4;
                    ELSE IF ( 1 < x4 <= 2 )
                            THEN gob_swept*(2-x4) + gob_vox*(x4-1);
                            ELSE IF ( 2 < x4 <= 3 )
                                    THEN gob_vox*(3-x4) + gob_blobby*(x4-2);
                                    ELSE IF ( 3 < x4 <= 4 )
                                            THEN gob_blobby*(4-x4) + gob_csg*(x4-3);
```

{ "Geometric types" section }

```
x1 : X;
x2 : Y;
x3 : Z;
x4 : T;
```

{ "Environment" section }

```
{ modeling space }
x1min = -11.;
x1max = 11.;
x2min = -11.;
x2max = 11.;
x3min = -11;
x3max = 11.;
x4min = 0.;
x4max = 4.;
{ incremental interval for x4 with geometric type "T" }
x4_delta = 0.5;
{ "Parameters" }
r1 = 4.;
r2 = 6.;
d1 = 5.;
r3 = 3.5;
e1 = -8.;
e2 = 2.5;
b  = -0.07;
a = [1.5, 1., 1., 0.8, 0.5, 0.3, 0.4, 0.35 ];
px = [0., 6., -4., -5., 5., 6., 5., 8. ];
py = [0., 6., -4., 3., -5., -7., -6., -9. ];
pz = [0., 6., -4., 7., 5., 9., -3., -4. ];
```

Captions

Fig.1. A segment in E^1 described as intersection of two rays. The intersection operation is defined by different R-functions: (a) α=1, (b) α=0, (c) m=1.

Fig.2. Constructing of "CSG" solid. Three initial solids "C", "S", and "G" are defined as a union of blocks. The final solid is defined as an intersection ("C"∩"S")∩"G".

Fig.3. The contour map of the intersection of two 2D halfspaces $x\geq0$ and $y\geq0$ represented by the R-function with α=0. The contour with f=0 is drawn with the bold line.

Fig.4a. Blending followed by the set-theoretic subtraction (a cylindrical hole).

Fig.4b. Negative and positive offset solids obtained by the iso-valued offsetting operation.

Fig.4c. Rounding convex vertices of a 2D solid by the blending based on the constant-radius offsetting operations.

Fig.5a. Twisting a constructive solid with the bijective mapping.

Fig.5b. A union of three and a projection of the object to a plane orthogonal to an axes of one of tori.

Fig.5c. The application of the Cartesian product and bijective mapping: a 3D solid defined by the rotational sweeping.

Fig.5d. Several steps of time dependent metamorphosis between 3D solids.

Fig.6. Metamorphosis modeling program in high-level geometric language

Fig.7. Frames of metamorphosis process between "key volumes" reflecting different representational styles: constructive geometry, sweeping, soft objects and voxel-based objects.

Fig.8 **a.** The body and the bottom of a wine glass to be connected with an aesthetic blend defined by the hand-drawn stroke;

b. The result of blending union with the estimated parameters.

Fig.9. Simulation of colliding particles sticking to each other.

Fig 10. Application of the set-theoretic operations for NC machining:

a. The set-theoretic subtraction between two moving solids; **b.** The object cut achieved as a result of the subtraction of the solid swept by a linearly moving cutter from the rotating workpiece.

Fig.11. Two functionally represented cross-sections and a solid reconstructed using metamorphosis operation.

Fig.12a. Metamorphosis between a vase and solid noise produces a noisy vase.

Fig.12b. Furry object defined by the offsetting and intersection with fur strands described with solid noise.