Function Representation for Sweeping by a Moving Solid

A.Sourin and A.Pasko

School of Applied Science
Nanyang Technological University
Nanyang Avenue
Singapore 2263

Laboratory of Shape Modeling
The University of Aizu
Aizu-Wakamatsu City
Fukushima, 965 Japan

ABSTRACT

This paper studies a function representation of point sets swept by moving solids. The original solid-generator is defined by an inequality $f(x,y,z,t)\geq 0$ where x,y,z are Cartesian coordinates and t is treated as the time. This definition allows us to include solids which change their shapes in time. Constructive solids can be used as generators also when described by *R*-functions. The trajectory of the generator can be defined in parametric form as movement of its local coordinate system. In the paper, we did it with superposition of time-dependent affine transformations. To get the function representation $F(x,y,z)\geq 0$ of the swept solid we apply the concept of envelope used before basically for boundary represented objects. We have reduced the problem of swept solid description to global extremum search by t variable. The algorithm of procedural swept solid modeling is discussed.

The benefit of our model is that it is applied not only for visualization but allows one to use the swept solid as an argument for another operations. For example, the swept solid can be intersected with another ones that is useful for the implementation of such operations as cutting and drilling. Ordinary texture mapping and hypertexturing can also be applied to it. The possibility of using a functionally defined generator with the variable shape allows us to achieve a complexity of the swept solids which was hardly possible before.

KEYWORDS
Solid Modelling, Sweeping, Implicit Functions, R-functions, Function Representation.

1 Introduction

Sweeping is one of the constructive techniques of solid modeling that creates new rigid solids which are not simple Boolean constructions and combinations of solid components. The full integration of this operation in solid modelers has not yet been achieved. This paper is an attempt to treat solids modeled with set-theoretic operations and with sweeping on the general base of representation by a real function of three variables. It provides a way for solving such long standing problems as sweeping by a CSG-solid, self-intersections and set operations on sweeps.

1.1 Definition

Two types of sweeping are usually distinguished. They are sweeping by a planar contour or "2D solid" (called also extrusion) and sweeping by a moving solid. Last one is the subject of this paper.

Sweeping can be generally defined as follows:

$$S = \bigcup_{t=t_1}^{t_2} G(t) \tag{1}$$

where S is a swept solid, $G(t)$ is a moving solid or a generator, $\bigcup$ is union operation of all instances of $G(t)$ for sweep parameter t in the interval $[t_1,t_2]$. In other words, a given point (x,y,z) is incident to S if there is t_0 in $[t_1,t_2]$ such as the point (x,y,z) is incident to $G(t_0)$.

1.2 Applications

The modeling of sweeping by moving solids can be important in many application areas. The examples of these applications are

- NC machining verification;
- Mould design with set-theoretic operations on swept solids;
- Motion planning of manipulators in robotics to avoid collisions with objects of the environment;
- Blending a solid with a desired surface by rolling a ball;

1.3 Methods

Among different methods one can distinguish between approximation of sweep solid and generation of sweep boundary.

In approximation, the generator can be converted to approximate ray-representation as it was suggested in [1]. Then, the swept solid is modeled as a union of a series of generator instances. If it is possible to represent a generator by depth data or "dexels", the swept solid can be approximated by the union of the solids swept out by each of these dexels.

Methods of sweep boundary generation often depend on particular shape of the generator and types of director. The detailed survey of polyhedral boundary generation for sweeping by polyhedral solids can be found in [2]. Canal surfaces swept by a sphere moving along a curve have been studied by Monge in the last century. Several authors have described boundary generation and rendering solids swept by special generators (sphere and cylinder) [3-6]

Two general approaches to define the boundary of the swept solid are Minkowski sums (see [5, 6, 1] and the theory of envelopes (see [3, 7]). If motion of the generator can be expressed in terms of one parameter (motion along a parametric curve, for example), the theory of envelopes seems to be more suitable.

Direct inclusion of a sweep operation in a CSG-tree is also considered (see, for example, [8]). The point membership classification algorithm based on an inverse sweep provides so-called sweep-CSG representation. However, a problem of detecting sweep parameter value for a given point remains unsolved for the general motion of the generator.

1.4 Problems

The main problems of swept solid modeling can be summarized as follows:

- General type motion of the generator;
- Piecewise smooth trajectory;
- Arbitrary and variable shape of a generator;
- Self-intersections of a swept solid;
- Sweeping by a CSG-solid;
- Set-theoretic operations on swept solids.

2 Function representation of solids

2.1 Static solids

In further considerations, we will represent solid objects as closed sets of 3D points with the following form of the defining function

G: $f(x,y,z)\geq 0$,

where G is some solid object, f is some real continuous functions of points coordinates and $f>0$ for points inside the solid, $f=0$ for points on the solid's border, and $f<0$ for points outside the solid. Such representation is well-known in CAD, computer graphics and animation. To transform or deform the solid we just change its defining function. In other words, we consider these defining functions and operations on these functions to model geometric objects and operations on these objects. Defining functions of complex solids can be created from a finite set of solid primitives or given defining functions with set-theoretic operations by CSG-like scheme. The exact analytical definitions of set-theoretic operations have been proposed by Rvachev in the theory of *R*-functions and applied for solving problems of mathematical physics. Applications of *R*-functions in solid modeling are described in [9, 10]. In the following considerations, $f_1 \wedge f_2$ stands for *R*-intersection, $f_1 \vee f_2$ means *R*-union, $f_1 \setminus f_2$ stands for *R*-subtraction and $-f$ for *R*-complement.
For example, if geometric object G_1 is defined as $f_1(x,y,z)\geq 0$ and geometric object G_2 is defined as $f_2(x,y,z)\geq 0$ then the following *R*-functions are the most useful in practice.

Intersection

$G_3 = G_1 \cap G_2$: $f_3 = f_1 \wedge f_2 = f_1 + f_2 - \sqrt{f_1^2 + f_2^2}$

Union

$G_3 = G_1 \cup G_2$: $f_3 = f_1 \vee f_2 = f_1 + f_2 + \sqrt{f_1^2 + f_2^2}$

Complement

$G_3 = \neg G_1$: $f_3 = -f_1$

Subtraction

$G_3 = G_1 \setminus G_2$: $f_3 = f_1 \setminus f_2 = f_1 - f_2 - \sqrt{f_1^2 + f_2^2}$

These *R*-functions have C^1 discontinuity only in points where both arguments are equal to zero.

Besides these set-theoretic operations, different classes of other operations can be defined for functionally represented solids. They are common bijective mapping, affine and projection transformations, offsetting, blending, and metamorphosis. In this paper we applied our functional approach to sweeping by a moving solid.

2.2 Moving solids

Considering defining functions of solids, we can easy make these solids moving introducing the forth variable - time t. Thus, general defining function of moving solid is $f(x,y,z,t)\geq 0$. One can propose plenty of methods how to get this function of four variables for the same moving solid. It could be straightforward function deriving, or using of parametric functions $x(t)$, $y(t)$, $z(t)$, or, at last, using of superposition of functions. The superposition of functions is particularly useful if the moving solid is considered as a static one being transformed by some time-dependent affine transformations. For better clarity of further consideration, we shall define several terms. We will call any time-dependent geometric object defined by the defining function $f(x,y,z,t)\geq 0$ a moving object *P*. It can be some rigid solid or variable-shape solid moving by path. For every particular $t=t_{const}$, we will name object $G=P(t)$ a sample of the moving object *P*. The simplest time-dependent object *P* can be defined as some static geometric object *G* defined on some time interval $[t_1, t_2]$. At the moment t_1, the object *G* with the defining function f_1 appears in 3D space and exists until the moment t_2. During this interval $[t_1,t_2]$ the samples of the moving object *P* are always *G=const*. Mathematically it is expressed as

$$G:\ f_1(x, y, z) \geq 0$$
$$P:\ f_2(x, y, z, t) = f_1(x, y, z) \wedge ((t - t_1) \wedge (t_2 - t)) \geq 0$$

where four-dimensional object *P* is a Cartesian product of 3D points and a time-segment. Note that Cartesian product is achieved as *R*-intersection of $f(x,y,z)\geq 0$ and time segment $(t - t_1) \wedge (t_2 - t)$. Recalling that we might want to define a bounding volume for 3D points, the more general form of the representation can be used

$$P:\ f_2(x, y, z, t) = f_1(x, y, z) \wedge ((t - t_1) \wedge (t_2 - t)) \wedge ((x - x_1) \wedge (x_2 - x)) \wedge ((y - y_1) \wedge (y_2 - y)) \wedge ((z - z_1) \wedge (z_2 - z)) \geq 0$$

Actually, in this formula, the four-dimensional bounding hyper-box for the moving object *P* is used.

A complex moving object can be defined as

$$P:\ f_2(x,y,z,t) = f_1(x,y,z,t) \wedge hbox(x_1,y_1,z_1,t_1,x_2,y_2,z_2,t_2) \geq 0$$

where $f_1(x,y,z,t)$ itself is function of t and can be derived either straightforward, or as $f_1(x(t),y(t),z(t))$, or as $f_1(A_n^{-1}(\ldots A_2^{-1}(A_1^{-1}(x,y,z))\ldots)$, where A_i^{-1} is inverse of some time-dependent affine transformation $A_i(t)$: $[x'\ y'\ z'\ 1]=[x\ y\ z\ 1][\{a[4,4](t)\}]$. The hyper-box *hbox* is defined as

$$hbox(x_1,y_1,z_1,t_1,x_2,y_2,z_2,t_2) = ((t-t_1)\wedge(t_2-t))\wedge((x-x_1)\wedge(x_2-x))\wedge((y-y_1)\wedge(y_2-y))\wedge((z-z_1)\wedge(z_2-z)) \geq 0$$

Thus, function

$$P_1:\ f_1(x,y,z,t) = 1-\left(\frac{x}{a}\right)^2-\left(\frac{y}{b}\right)^2-\left(\frac{z}{c}\right)^2-\left(\frac{t}{d}\right)^2 \geq 0$$

which actually defines a hyperellipsoid in 4D space, can be considered as a defining function for the moving solid which is 3D ellipsoid subsequently increasing and decreasing in size during t=[-d,d]. The speed of this change of shape is nonlinear. Intersecting this function with, for example, time segment [0,d/2]

$$P_2:\ f_2(x,y,z,t) = f_1(x,y,z,t)\wedge((t)\wedge(\frac{d}{2}-t)) \geq 0,$$

one can define a part of this process, namely, the ellipsoid decreasing in size during $t\in$[0,d/2].
Parametric definitions and time-dependent affine transformations seem to be more suitable for defining movement by path and simple deformations. For example, the above mentioned moving ellipsoid might be spinned and moved horizontally as follows

$$P_3 = T_1(T_2(P_2)):\ f_1(f_{T2}^{-1}(f_{T1}^{-1}(x,y,z,t)))\wedge((t)\wedge(\frac{d}{2}-t)) \geq 0,$$

where f_{T1}^{-1} and f_{T2}^{-1} are inverse of time-dependent affine transformations

T_1=*Translation*(0, 0, $dz(t-t_1)/(t_2-t_1)$) and
T_2=*Rotation_about_OZ*($2\pi n(t-t_3)/(t_4-t_3)$)

We do not stipulate the user and allow him to choose any possible form for the representation of $f(x,y,z,t)$. But often, it is more convenient to use time-dependent affine transformations as a very familiar mathematical representation. And besides, these affine transformations can also be applied to the forth coordinate t, both in $f(x,y,z,t)$ and in the bounding hyper-box. For example, time translation can be treated as bringing forward and postponing, time scaling - as acceleration and deceleration. More unusual operations with hardly predictable results can be created if we consider, for example, 4D rotation about 3D plane or scaling relative to some 4D point.

3 Function representation of sweeps

Let us consider now how to represent a sweep in the functional form. Suppose that we have defined somehow the moving object *P* which is some rigid solid moving by path. The sweep created by this moving solid is actually projection of moving object *P* defined in four-dimensional space to 3D Cartesian coordinate space (x,y,z). Because we cannot implement this projection as simple elimination of the coordinate t, we define it as follows:

$$G:\ P(t_1)\cup P(t_1+dt)\cup P(t_1+2dt)\ldots P(t_2)$$

where $dt\to 0$ and $\cup$ is set-theoretic union.

If object P is defined by the defining function $f_1(x,y,z,t)\geq 0$, the function representation of the sweep can be given as the following procedure:

<Sweep>
$f(x,y,z)$
{
$Fs=f_1(x,y,z,t_1)$;
for every t from $[t_1+dt,t_2]$ with increment $dt\to 0$ do:
{ *Evaluate* $Fp=f_1(x,y,z,t)$;
$Fs=Fs \vee Fp$;}
}

Just for visualization, we can apply another procedure which uses bounding hyper-box, provided it is defined in f_1.

<Sweep>
$f(x,y,z)$
{
Evaluate $Fs=hbox(x_1,y_1,z_1,x_2,y_2,z_2)$;
if $Fs<0$ *then return*;
$Fs=f_1(x,y,z,t_1)$; (2)
for every t from $[t_1+dt,t_2]$ with increment $dt\to 0$ do:
{ *Evaluate* $Fp=f_1(x,y,z,t)$;
$Fs=Fs \vee Fp$;
if $Fs\geq 0$ then break;}
}

This procedure defines the discontinuous function that is suitable only for zero root search using binary subdivision method. If Newton iterations are involved, this procedure is not appropriate.

In Fig. 1a, the example of the implementation of such a method is given. Here we met the problem of aliasing which depends on sampling or incremental nature of the algorithm. The sweep has jagged shape. It looks like made with chisel. Like in 2D rasterization, we have to use some antialiasing methods. We can either increase the number of samples reducing dt that obviously makes these calculations very time consuming, or use some antialiasing algorithms when unifying these samples. A naive approach to deal with this problem is to apply blending operation to neighbouring samples of a moving solid. We applied the following blending R-union operation:

$$Fs \vee Fp = Fs + Fp + \sqrt{Fs^2 + Fp^2} + \frac{a_0}{1+\left(\frac{Fs}{a_1}\right)^2 + \left(\frac{Fp}{a_2}\right)^2}$$

where Fs and Fp are functions from (2), and a_0, a_1, a_2 are some parameters. This definition provides highly intuitive shape control of added material, subtracted material, and variable radius blends. As it can be seen in Fig. 1b, the result is highly dependent on the curvature of the trajectory, if a uniform set of blending parameters is applied to all pairs of samples.

Another function representation of the sweep created by a moving solid can be defined as union of two solid samples in initial and final positions and an envelope swept by the moving solid. This concept of envelope borrowed from classical differential geometry was used by several authors to derive analytical representations of envelopes swept by simple geometric objects.

The swept object can be described as

$$f(x, y, z) = f(x, y, z, t_1) \vee f(x, y, z, t_2) \vee E(x, y, z), \quad (3)$$

where E is the description of the new object swept during the motion of the generator with the defining function f. This new object is called an envelope. The surface of the envelope has the following description

$$\begin{cases} f(x, y, z, t) = 0 \\ \dfrac{df(x, y, z, t)}{dt} = 0 \end{cases} \quad (4)$$

Commonly, the second equation has to be solved for t in terms of (x,y,z) and then this t should be substituted in the first equation to reduce the dimension of variables and to yield the description of the envelope. But this way is possible

only if the function derivative is expressed analytically. In our case, when the defining function can be defined procedurally, this approach cannot be applied.

To find the algorithm of the envelope defining function evaluation, let us reformulate the description (4). Zero value of the derivative in (4) indicates that for given (x,y,z) the function f takes an extremal value on the interval $[t_1,t_2]$ on the envelope surface. Taking into account negative function values outside the object, it can be stated that

$$E(x,y,z) = f(x,y,z,t_0) \wedge ((t-t_1) \wedge (t_2-t)) \quad \text{where } t_0 = \arg\max(f(x,y,z,t)), \quad t \in [t_1,t_2] \tag{5}$$

Note that function $f(x,y,z,t)$ can have several local extrema, for example, when the envelope has self-intersections. Therefore, the problem of global function maximum search has to be solved in a general case.

The following numerical algorithm for t_0 search can be proposed.

1) For given (x, y, z)

evaluate $f_i(t_i),\ t_i \in [t_1,t_2],\ dt = t_{i+1} - t_i$;

2) Find t_k with maximal f_i value;

3) Interpolate function value using Newton's interpolation

$$\hat{f}(p) = f_{k-1} + a \cdot p + b \cdot p \cdot \frac{p-2}{2} \qquad \text{with } p \in [0,2] \text{ and}$$

$$a = f_k - f_{k-1},\ \ b = (f_{k+1} - f_k) - (f_k - f_{k-1})$$

4) Define $\hat{f}'(p) = b \cdot p + \dfrac{2a-b}{2}$

5) Assuming $\hat{f}'(p) = 0$

define $t_0 = t_1 + (k-2) \cdot dt + \left(\dfrac{b-2a}{2b}\right) \cdot dt$

If $t_k=t_1$ or $t_k=t_2$, the interpolation interval has to be changed or extra information has to be added on f values for $t=t_1-dt$ or $t=t_2+dt$. After t_0 is found, the envelope defining function can be evaluated by (5) and the whole sweep defining function can be evaluated by (3). Note that the function (3) represents a swept solid rather than its boundary.

The example of sweep created by this method is given in Fig. 1c. It is obvious that quality of this object is much better than in Fig. 1a. Note that calculating time spent for this algorithm is the same as for the previous one. Fig. 2 gives another illustration of the ability to define complex self-intersected motion of an arbitrary generator. The trajectory of the generator is a Lissajous curve - the same as for Fig. 1. In turn, the generator itself is moving object spinning about a vertical axes and created with *R*-union of two solids.

4 Implementation and problems

In Fig. 3(a-f), we show some results of application of this method to solving NC-machining verification problem. Both, workpiece and cutter can be complex moving objects defined with some time-dependent affine transformations A_1, A_2 and set-theoretic operations. The general approach used here is set-theoretic subtraction of the swept solid from another solid object done by the following algorithm:

< Subtraction of the moving cutter from the moving workpiece >

$f(x,y,z,t_i)$

do:

Evaluate $f_1(x,y,z,t)$; /* Sample of the moving workpiece at t */

Evaluate $f_2(x,y,z,t_1) = f_2(A_1^{-1}(A_2(x,y,z,t_1)))$; /* Sample of the moving cutter at t_1 in the workpiece coordinate system */

Evaluate $f_2(x,y,z,t_i) = f_2(A_1^{-1}(A_2(x,y,z,t_i)))$; /* Sample of the moving cutter at t_i in the workpiece coordinate system */

Evaluate $E(x,y,z)$ *of* $f_2(A_1^{-1}(A_2(x,y,z,t)))$ *for* $[t_1,t_i]$ *interval*; /* Envelope created by the moving cutter as seen from the workpiece coordinate system */

$f_3 = f_2(x,y,z,t_1) \vee f_2(x,y,z,t) \vee E(x,y,z)$;

$f = f_1 \setminus f_3$

enddo;

In this algorithm, the set-theoretic operation is being performed in the workpiece coordinate system. This is the operation over the stationary workpiece and the object swept by the moving cutter as seen from the workpiece coordinate system. In our example - over the cube and the sphere moving around the cube.

A solid, depicted in Fig. 3a, is the result of set-theoretic subtraction of the sphere fed horizontally from the spinning cube. A bolt, in Fig. 3b, is another example of subtraction of the swept solid defined by moving cutter from the CSG solid. Our model allows us to simulate both rough and finish machining simply varying speeds of the workpiece and cutter. Thus, in Fig. 3(c-f), 'feed marks' are results of slow rotational speed of the four- and two-flute side mills. With increasing this speed, the surface of the workpiece is getting smother.

In Fig. 4, we show quite complex sweep created by two moving solids. Below, the pseudocode of the simulation procedure is given.

/* Shape definition by CSG scheme */

$G1 =$ '*cup*';

$G2 =$ '*spoon*';

/* Making moving objects P_1 and P_2 assigning time intervals $[t_0, t_3]$ */

$P_1 = G_1 \wedge ((t - t_0) \wedge (t_3 - t))$;

$P_2 = G_2 \wedge ((t - t_0) \wedge (t_3 - t))$;

/* Movement definition for "*moving cup*". Falling down from t_1 till t_3.*/

$P_1 = \underset{t1-t2}{T_1}(P_1)$;

/* Movement definition for "*moving spoon*". Falling down together with the cup from t_1 till t_2.
Additional component of movement from t_2 till t_3. */

$P_2 = \underset{t2-t3}{T_2}(\underset{t1-t3}{T_1}(P_2))$;

/* Making a complex solid object "*moving cup and spoon*". */

$P3 = T3(P1 \; U \; P2)$;

/* Making a sweep for $[t_1, t_3]$ */

$P_4 = P_3(t_1) \cup P_3(t_2) \cup Envel(P_3, t_1, t_3)$;

These solids are a cup and a spoon. Initially they both calmly stand on the table and spoon is in the cup. Then (at t_1) they begin falling down the table, soon (at t_2) the spoon is leaving the cup and they continue their movement separately (until t_3). Modeling this application we tried to solve the common problems of sweeping mentioned in subsection 1.4. The cup and spoon were created as constructive solids with set-theoretic operations. The defining functions were obtained by applying *R*-functions in a procedural manner. Then, these two solids are made time-dependent. It is provided by the separate descriptions of cup and spoon movement. Then the moving cup and spoon are combined in a single solid by set-theoretic union. Two moving parts of this solid have common points at the first moment. Then, at $t=t_2$, the spoon starts to fall out of the cup. Thus, two disconnected components, still described by a single defining function, appear. This complex moving solid consisting of two moving parts is used as a generator for the sweep P_4 eventually defined. This sweep is defined functionally and can be used in other operations closed on our model. In other words, the sweep and its generator belong to the same class of objects defined functionally. We also successfully solved the problem of self-intersection. Points swept by the cup handle belong also to spoon instances with different t value. This situation is usually classified as a "self-intersection problem". In our approach it causes existence of more than one local extremum of the defining function by t variable in a given point. A procedure of global extremum search has to be applied. Note that all above mention problems have not received a general solution before.

Talking about drawbacks of the method discussed, we just point out the main of them. This is huge processing time. Actually, sweeping is time-consuming whatever method you choose and this is the common drawback of sweeping. In the above algorithm, the most of time is spent for function extremum searching and reducing this time we can achieve much better overall performance of the algorithm. The methods which can be used are

- nonuniform increment in searching extremum
- gradient search
- genetic optimization algorithm

In our implementation, we used time-dependent affine transformations for defining moving solids. We precalculated and stored in the data structure all matrices of transformations used in calculation that provided us significant acceleration. Using of bounding boxes gave us another improvement. Note, that we didn't try to use parallel computation yet, but nature of our model assume that this parallelism can be easily achieved. It will be done in due course.

5 Conclusion and future work

In this work, we proposed and studied the functionally based representation for sweeping by moving solid. Several major problems can be solved with this approach.

1. Arbitrary variable-shape solids can be defined as generators with procedural defining functions.
2. Solid generators can be created by CSG-like scheme.
3. Sweeps of arbitrary complex solids over arbitrary movements with self-intersections can be handled.

We point out that the above tasks were not solved within one single model before.

The further problems to be solved here are

- Detecting characteristic points of a trajectory curve and using them in the described maximum search procedure;
- Comparison of different maximum search methods: gradient method, stochastic and genetic algorithms;
- Generalization of the proposed approach on the case of a moving solid that keeps continually in contact with one or several surfaces.

References

1. Menon J., Marisa R.J., Zagajac J., "More powerful solid modeling trough ray representations", *IEEE Computer Graphics and Applications*, vol.14, No.3, 1994, pp.22-35.

2. Sambandan K., "Geometry generated by sweeps of polygons and polyhedra", Technical Report No.66, Cornell Univesity, Ithaca, August 1990, 156 p.

3. Wang W.P., Wang K.K., "Geometric modeling for swept volume of moving solids", *IEEE Computer Graphics and Applications*, vol.6, No.12, 1986, pp.8-17.

4. van Wijk J.J., "Ray tracing of objects defined by sweeping a sphere", *Computers and Graphics*, vol.9, No.3, 1985, pp.283-290.

5. Ghosh P.K., "A unified computational framework for Minkowski operations", *Computers and Graphics*, vol.17, No.4, 1993, pp.357-378.

6. Middleditch A.E., "Ray casting set-theoretic rolling sphere blends", Computer-aided Surface Geometry and Design, The Mathematics of Surfaces IV, A.Bowyer (Ed.), Clarendon Press, Oxford, 1994, pp.261-280.

7. Martin R.R., Stephenson P.C., "Sweeping of three-dimensional objects", *Computer-Aided Design*, vol.22, No.4, 1990, pp.223-234.

8. Hui K.C., "Solid modelling with sweep-CSG representation", CSG 94 Set-theoretic Solid Modelling: Techniques and Applications, Information Geometers, Winchester, 1994, pp.119-131.

9. Shapiro V., "Real functions for representation of rigid solids", *Computer Aided Geometric Design*, vol.11, No.2, 1994, pp.153-175.

10. Pasko, A.A. et al., "Function Representation in Geometric Modeling: Concepts, Implementation and Applications", To appear in *The Visual Computer*.

Captions

Figure 1
(a) Sweep created with *R*-union of many samples
(b) Sweep created with blending *R*-union of many samples
(c) Sweep created with functionally defined envelope

Figure 2
(a) Samples of complex moving generator
(b) Swept solid

Figure 3
NC-machining applications

Figure 4
Sweeping by the complex moving solid with time-dependent shape and self-intersection.
(a) Two samples of the generator in the initial and final positions
(b) Union of 8 samples of the moving generator
(c) Solid object swept by the complex generator

Authors' Biographies

Alexei I. Sourin received his M.Sc. and Ph.D. degrees in Computer Science from Moscow Engineering Physics Institute (MEPI), Russia, in 1983 and 1988 respectively. For the period of 1983 - 1993 he was a researcher at MEPI. Currently, he is a lecturer at School of Applied Science, Nanyang Technological University, Singapore. His interests include solid modelling with real functions, computer animation, scientific visualization.

Address:
School of Applied Science,
Nanyang Technological University,
Nanyang Avenue, Singapore 2264.
E-mail: assourin@ntuvax.ntu.ac.sg
Fax: +65-792-6559

Alexander A. Pasko received M.Sc. and Ph.D. degrees in Computer Science from Moscow Engineering Physics Institute (MEPI, Russia) in 1983 and 1988. He was a researcher at MEPI from 1983 to 1992. Currently he is an assistant professor at Shape Modeling Laboratory, Computer Software Department, University of Aizu, Japan. His research interests include solid modeling, animation, multidimensional visualization and experimental data processing. He is a member of ACM, IEEE and Eurographics Association.

Address:
Shape Modeling Lab.,
University of Aizu, Aizu-Wakamatsu City,
Fukushima 965-80, Japan.
E-mail: pasko@u-aizu.ac.jp
Fax: +81-242-372728