**UDK 621.324**

S. I. VYATKIN
Institute of Automation and Electrometry SB RAS
A. N. ROMANYUK, B.L. VOIT
Vinnytsia National Technical University

# PERTURBATION FUNCTIONS AND OPERATIONS IN GEOMETRIC MODELING

*The free forms based on the analytical perturbation functions have an advantage of spline representation of surfaces, that is, a high degree of smoothness, and an advantage of arbitrary form for a small number of perturbation functions. We have investigated geometric operations on functionally defined objects on the basis of the perturbation functions. We have analyzed the collision detection algorithm by means of recursive object space subdivision. The free-form representation created by mean of the analytical perturbation functions have the following advantages: fewer surfaces for mapping curvilinear objects, short database description, fewer operation for geometric transformations and data transfer, simple animation and deformation of objects and surfaces. For shape creating we propose a set of algorithms and software based on function-defined surfaces that perform an interactive rate and enable intuitive operations. Interactive modification of the function model with fast visualization allows us to provide both the interactivity and any required level of detail leading to a photo-realistic appearance of the resulting shapes. Our investigation in the volume-oriented visualization technology has made it possible to reveal some advantages in both the scene representation technique and the rendering algorithm. The main merits of our approach are the following: reduced number of surfaces for describing curvilinear objects (representation of objects by free-form surfaces reduces 100 times and more the database description compared with their representation by polygons); efficiency of the masking rendering technique combining simple computation with fast search and discard of spaces out of the scene objects; the possibility to process voxel arrays bounded by freeform surfaces; reduction of the load on the geometry processor and decrease of data flow from it to the raster subsystem; simple animation and morphing of scenes. The proposed visualization algorithm along with the possibility to visualize arbitrary surfaces of free-forms and inhomogeneous volume spaces offers a wide scope of application. The free-form representation has a wide spectrum of applications (interactive graphics systems for visualizing functionally defined objects, CAD 3-D simulation systems, 3-D web visualization, prototyping, etc.). More effective tools for designing, manipulating, and deforming free-form 3-D shapes are needed in CAD, animation, and virtual reality applications.*

*Keywords: perturbation functions, geometric operations, collision detection, volume-oriented visualization*

С.І. ВЯТКІН
Інститут автоматики та електрометрії СБ РАС
О.Н. РОМАНЮК, Б.Л. ВОЙТ
Вінницький національний технічний університет

## ФУНКЦІЇ ПЕРЕРУВАННЯ І ОПЕРАЦІЇ В ГЕОМЕТРІЧНЕ МОДЕЛЮВАННЯ

*Вільні форми, засновані на аналітичних функціях збурень, мають перевагу сплайн-представлення поверхонь, тобто високий ступінь гладкості і перевага довільної форми для невеликої кількості функцій збурень. Ми дослідили геометричні операції на функціонально визначених об'єктах на основі функцій збурень. Ми проаналізували алгоритм виявлення зіткнень за допомогою рекурсивного підрозділу космічного об'єкта.*

*Ключові слова: функції збурень, геометричні операції, виявлення зіткнень, об'ємно-орієнтована візуалізація*

## Introduction

Several representations of geometric objects are currently used in computer graphics. Each of the objects, according to its properties, is used in different fields, beginning from 3D simulation and CAD systems up to real-time visualization systems.

The polygonal surface representation is a piecewise linear interpolation of a surface. Its merit is a simple representation and universal application because the piecewise linear representation exists for any surface. We should mention the insignificant computational expenses required for visualization and geometric transformations. The drawback is large database for storing the information on the surface geometry. Highly detailed models (multiresolution geometric models), e.g., antique sculptures, subjected to computer reconstruction have hundreds of millions of triangles.

The spline representation of surfaces [1], along with analytical representation, is used in AutoCAD and 3D Studio systems. It is characterized by a highly accurate representation of 2D and 3D objects.

The functional representation [2] describes most accurately the object geometry and has the smallest size of the required data. Procedures of functional representation demonstrate compact and flexible representation of surfaces and objects that are results of logical operations on volumes.

New techniques for specifying free forms without their approximation by polygons or spline-patches are considered. We suggest expanding the notion of primitives and making it possible to process them by easy and effective method without approximation by polygons. A method to display curved surfaces allows obtaining picture quality which cannot be achieved by the traditional means (even with great number of polygons) and is described below.

The geometric concept of virtual environment modeling using function-based objects can be described as an algebraic system (1):

$$(M, \Phi, W), \tag{1}$$

where M is the set of geometric objects, Φ is the set of geometric operations, and W is the set of relations on the set of objects. Geometric objects are considered as closed subsets of n-dimensional Euclidean space En with the definition(2):

$$f(x1, x2, \ldots x_n) \geq 0, \tag{2}$$

where $f$ is a real continuous function defined on En.

A functionally defined object is completely defined by means of the real-valued describing function of three variables (x1, x2, x3) in the form of F(X) ≥ 0, then the objects are treated as closed subsets of the Euclidean space En, defined by the describing function F(X) ≥ 0, where F is the continuous real-valued function and X= (x1, x2, x3) is the point in En, defined by the coordinate variables. Here F(X) > 0 defines points inside the object, F(X) = 0 defines points on the boundary, and F(X) < 0 defines points that lie outside and do not belong to the object.

It is possible to describe complex geometry forms by specifying surface deviation function (of second order) in addition to surface basic function of second order. Generally a function F(x,y,z) specifies surface of second order that is quadric (3),

$$F(x,y,z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + A_{13}xz + A_{23}yz + A_{14}x + A_{24}y + A_{34}z + A_{44} \geq 0, \tag{3}$$

where x, y and z are spatial variables.

**Inclusion relation**

This relation is described as $G_2 \subset G_1$ and means that the object $G_2$ is a subset of $G_1$. If is a point P, the $G_2$ relation can be described by the following bivalued predicate (4):

$$S_2(P,G_1) = \begin{cases} 0, if \ f_1(x,y,z) < 0 \ for \ P \notin G_1 \\ 1, if \ f_{1i}(x,y,z) \geq 0 \ for \ P \in G_1 \end{cases} \tag{4}$$

**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(5):

$$S_3(P,G_1) = \begin{cases} 0, if \ f_1(x,y,z) < 0 \ for \ P \notin G_1 \\ 1, if \ f_{1i}(x,y,z) = 0 \ for \ P \in bG_1 \\ 2, if \ f_1(x,y,z) > 0 \ for \ P \in iG_1 \end{cases} \tag{5}$$

This predicate can be correctly evaluated for $G_1$ without internal zeroes, internal points with $f_1(x,y,z) = 0$.

**Set theoretic operations**

Let the objects G1 and G2 be defined as $f1(X) \geq 0$ and $f2(X) \geq 0$. The binary operation (n=2) of the objects G1 and G2 means operation G3=Φ¡(G1,G2) with the definition (6)

$$f_3 = y\left( f_1(X), f_2(X)\right) \geq 0, \tag{6}$$

where ψ is the continuous real function of two variables. Let us dwell on the binary operations: set-theoretic operations.

For function-based objects on the bases of perturbation functions we propose the following. To create a complex scene, one should describe in it a certain number of primitives necessary for a concrete task. The rendered object with which the rendering algorithm interacts by means of query represents the whole 3D scene. Hence, the geometric model should allow designing of objects and their compositions of infinite complexity. This is primarily achieved by means of Boolean operations of uniting and intersection.

Let the objects $G_1 : f_1(x,y,z) \geq 0$ and $G_2 : f_2(x,y,z) \geq 0$, then $G_3 = G_1 \cup G_2$ is the union operation, $G_3 = G_1 \cap G_2$ is the intersection operation, $G_3 = G_1 \setminus G_2$ is the subtraction operation and $f_3 = y\ (f_1(x,y,z), f_2(x,y,z))$ [3].

**Intersection relation**

The intersection or interference and collision relation is defined by the bivalued predicate (7):

$$S_C(G_1,G_2) = \begin{cases} 0, if\ G_1 \cap G_2 = \varnothing \\ 1, if\ G_1 \cap G_2 \neq \varnothing \end{cases} \tag{7}$$

$$G_1 : f_1(x,y,z) \geq 0,\ G_2 : f_2(x,y,z) \geq 0$$

A function $f_3(x,y,z) = f_1(x,y,z) \,\&\, f_2(x,y,z)$ defining the result of the intersection can be used to evaluate $S_c$. It can be stated that $S_c = 0$ if $f_3(x,y,z) < 0$ for any point of $E^3$.

**Perturbation functions**

We propose describing complex geometric objects by specifying the function of deviation (an implicit second-order function) from the base surfaces [2]. The freeform is a composition of the base surface and the perturbation functions (8)

$$F'(x,y,z) = F(x,y,z) + \sum_{i=1}^{N} R_i(x,y,z) \tag{8}$$

where the perturbation function R(x, y, z) is found as follows (9)

$$R_i(x,y,z)=\begin{cases}Q_i^3(x,y,z), if\ Q_i(x,y,z)\geq 0\\ 0, if\ Q_i(x,y,z)<0\end{cases} \tag{9}$$

Herein, Q(x, y, z) is the perturbing quadric.

Since max[Q + R] ≤max[Q] + max[R], for estimating the maximum Q on some interval we have to calculate the maximum perturbation function on the same interval. The obtained surfaces are smooth, and creation of complex surface forms requires few perturbation functions.

Thus, the problem of object construction reduces to the problem of quadric surface deformation in a desired manner rather than to approximation by primitives (polygons or patches represented by B-spline surfaces). In addition, while solving the descriptive function in the form of inequality F(X) ≥ 0, we can visualize not only the surface but also the internal structure of the object.

**Collision detection**

An example of the relations is collision detection for objects. The binary relation is a set of the set $M^2=M\times M$. It may be defined as

$$S_i: M\times M \to I$$

Collision detection is a complicated problem solved in various computer programs [4]. This means that for each animation frame, one should test whether any two or more objects collided.

The ideal case is collision detection of any complexity between two arbitrary objects in the minimal time. Since the control of collisions between all pairs of objects is a resource-consuming process, such tests are usually done only for part of objects. The detection algorithm can be simplified prior to testing the presence of the given point (belonging to one of the objects), e.g., inside the cube confining the second object. The problem of simulating the behavior of interacting bodies having irregular shape arises in some applications such as dynamics of body collisions and celestial mechanics, molecular dynamics, graphics simulations for the problem of nano-assembly automation and its application in medicine using collective robotics [4], computer games and haptic interactions.

Particularly in calculating motions of many objects that move under changing constraints and frequently make collisions, one of the key issues of dynamic simulation methods is calculation of collision impulse between rigid bodies [5]. A fast algorithm for calculating contact force with friction by formulating the relation between force and relative acceleration as a linear complementary problem was equally demonstrated and this model was based on solving the linear complementary problem [6]. Baraff's algorithm has achieved great performance for real-time and interactive simulation of two-dimensional mechanisms with contact force, friction force and collision impulse, although friction impulse at collision was not completely covered in such a model. In geometric haptic rendering models, collision detection is not trivial to compute. One of the most popular collision detection algorithms in geometric haptic rendering is H-Collide [7]. It uses a hybrid hierarchy of spatial subdivision and OBB trees. The simplest algorithms for collision detection are based upon using bounding volumes and spatial decomposition techniques.

Examples of bounding volumes include bounding spheres, bounding boxes, convex polyhedrons. Examples of bounding boxes include axis-aligned bounding boxes and oriented bounding boxes. In work [8] authors used a bounding spheres hierarchy to detect collisions. Spatial decomposition techniques based on subdivision are used to solve the interference problem. Recursive subdivision is robust but computationally expensive. In particular, Hahn [9] used a subdivision based collision detection algorithm.

For curved objects, Herzen and etc. [10] have described an algorithm based on subdivision technique. A similar method using interval arithmetic and subdivision has been presented for collision detection by Duff [11]. However, for commonly used spline patches computing and representing the implicit representations is computationally expensive [12]. In [13], Pentland and Williams used implicit functions to represent shape and the property of the "inside-outside" functions for collision detection. But this algorithm has a drawback in terms of robustness, as it uses point samples. Thus, the most popular collision detection algorithms are extremely distance and extremely points (four nonlinear equations solving), testing sample points (accuracy of sampling using huge memory), interval methods (interval bounds on the output of functions with time-consuming). The detailed explanation of main problems is described in [11]. There are several problems exist. There are procedurally defined functions, time-dependent surfaces and surfaces of high complexity.

We propose the collision detection algorithm by means of recursive object space subdivision.

After calculation of the intersection, (10)

$$S_C(G_1,G_2)=\begin{cases}0, if\ G_1\cap G_2=\varnothing\\ 1, if\ G_1\cap G_2\neq\varnothing\end{cases} \tag{10}$$

i.e., application of the Boolean operation of intersection, the search for the contact point of collided objects is done by means of recursive subdivision of the object (model) space. Hence, it is sufficient to find at least one point (or more) belonging to intersection. Let we deal with the object-intersection that has the property of answering the request on intersection with a rectangular parallelepiped or a bar. The negative answer guarantees that the object is not intersected and has no common points belonging to the intersection is done by recursive subdivision of the space inside the cube defined by boundaries of ±1 along each coordinate. The center of the cube matches the origin of the model coordinate system M whereas the plane Z= −1 coincides with the screen plane. At the first step of recursion, the initial cube is subdivided into four smaller subcubes in the screen plane. At the stage of subdivision of space along the quaternary tree, 2-times compression and transfer by ±1 along two coordinates are performed. Assume, that domain of point search is a cube in which embed our object-intersection.

Then recursive subdivision of the domain applied: domain cuts by 2 planes, that perpendicular to the screen plane XY, into 4 bars. For each bar intersection test are executed. If the object intersects with given bar, then bar subdivides further. Otherwise, we exclude bar from subdivision. This corresponds with exclusion of the square areas in the screen, on which given bar (and therefore, object- intersection) are projected.

If in the equation of quadric Q(x, y, z) = 0 (4) the values of the variables x, y, z vary within the length [-1, 1], then

$$\max\left[|Q(x,y,z)-A44|\right] \le \max F = |A11|+|A22|++|A33|+|A12|+|A13|+|A23|+$$
$$+|A14|+|A24|+|A34|$$

We should note that if |A44| □ max[ |Q(x, y, z) – A44| ] □ maxF, then, probably, a point M0 = (x0, y0, z0) ( -1 < x0, y0, z0 < 1 ) exists such that Q(x0, y0, z0) = 0. If maxF < |A44|, then such points do not knowingly exist, and the sign of the coefficient A44 distinguishes location of the bar inside or outside with respect to the quadric surface Q=0 (if A44 ≥ 0, then the subbar is inside the quadric). Using results of this test, we perform subdivision of subbars that fall within the quadric completely or, probably, partially, and the knowingly external subbars are eliminated from processing. A test for intersection of subbars with freeform is somewhat different. For the basic quadric the test for intersection looks as follows (11):

$$if\ \left((A44+R)<0\right)\&\&(|A11|+|A22|+|A33|+|A12|+|A13|+ \\ +|A23|+|A14|+|A24|+|A34|<-(A44+R)) \quad (11)$$

then the subbar is outside.

Here R is the maximum perturbation function on the current interval; Aij are the coefficients of quadratic function. The following test is performed for the perturbation function (12):

$$if\ (|A11|+|A22|+|A33|+|A12|+|A13|+|A23|+ \\ +|A14|+|A24|+|A34|<|A44|) \quad (12)$$

then the subbar is outside of the range of definition of perturbation,

where Aij are the coefficients of the quadratic perturbation function and a value of R is additionally calculated and added to the basic function.

## Conclusion

The freeform representations created by means of the scalar and analytical perturbation functions have the following advantages: fewer surface for mapping curvilinear objects, short database description, fewer operations for geometric transformations and data transfer, simple animation and deformation of objects and surfaces, and a wide spectrum of applications (interactive graphics systems for visualizing function-based objects, CAD 3D simulation systems, 3D web visualization, etc.).

We may conclude that in the proposed function-based object collision detection algorithm, the collision is always detected and does not depend on the relative position of collided objects and parts of their surfaces, i.e., such an algorithm guarantees detection of the event, which has been proved both experimentally and theoretically, it is required to have equal number of levels of object space subdivision and, therefore, equal computation time. The object collision was detected in a constant time for collisions of different complexity and the time spread in the tests was below 1% of the given time.

## References


1. Robert E. B. and Richard F. R. COMPUTER AIDED GEOMETRIC DESIGN / Barnhill E.B., F. R. Richard // New York, San Francisco, London, 1974, Proceedings of a Conference Held at The University of Utah, Salt Lake City, Utah March — 1974. — P. 18-21.
2. Vyatkin S.I. Complex Surface Modeling Using Perturbation Functions / S.I Vyatkin // Optoelectronics, Instrumentation and Data Processing. — 2007. Vol. 43, No. 3, — P. 40-47.
3. Ricci A. A constructive geometry for computer graphics/ A. A. Ricci. // The Computer Journal — 1973. Vol. 16, No.2, — P. 157-160.
4. Lin M.C. Efficient Collision Detection for Animation and Robotics / Lin M.C. // PhD thesis, Dept. of Electrical Eng. and Computer Science, University of California, Berkeley, USA, 1993.
5. Baraff D. Analytical methods for dynamic simulation of non-penetrating rigid bodies / D. Baraff // Computer Graphics Proceedings, ACM SIGGRAPH. — 1989. Vol. 23, — P. 223-232,
6. Baraff D. Fast contact force computation for nonpenetrating rigid bodies / D. Baraff // in Computer Graphics Proceedings, Annual Conf. Series. ACM SIGGRAPH. — 1994. — P. 23-34.
7. Gregory, M. Lin, et al. "H-Collide: A Framework for Fast and Accurate Collision Detection for Haptic Interaction". IEEE Virtual Reality, 1999
8. Ruspini D. C., Kolarov K., and Knatib O. The haptic display of complex graphical environment/ D. C. Ruspini, K. Kolarov, O. Knatib // Proceedings of SIGGRAPH 97. — August 1997. Vol. 1, P. 295-301.
9. Hahn J. K. Realistic animation of rigid bodies / J. K. Hahn // ACM Computer Graphics. — 1988. Vol. 22, No. 4, — P. 299-308.
10. Herzen B.V., Barr A.H. and Zatz H.R. Geometric collisions for time-dependent parametric surfaces / B.V. Herzen, A.H. Barr, H.R. Zatz // ACM Computer Graphics, 24(4), August 1990.
11. Duff T. Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry / T. Duff // ACM Computer Graphics. — 1992. Vol. 26, No.2, — P. 131-139.
12. Hoffmann C. Geometric and solid modeling / C. Hoffmann // Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1989.
13. Pentland A. and Williams J. Good vibrations: modal dynamics for graphics and animation /
14. Pentland, J. Williams // ACM Computer Graphics. — 1990. Vol. 23, No. 3, — P. 185-192.


Рецензія/Peer review : 23.06.2017 р. Надрукована/Printed :10.10.2017 р.
Стаття рецензована редакційною колегією