Volume 14, (1995) number 4 pp. 181–188

COMPUTER GRAPHICS *forum*

Function Representation of Solids Reconstructed from Scattered Surface Points and Contours

Vladimir V. Savchenko, Alexander A. Pasko, Oleg G. Okunev and Tosiyasu L. Kunii

Department of Computer Software, The University of Aizu
Aza Kami-Iawase 90, Oaza Tsuruga, Ikki machi, Aizu-Wakamatsu City,
Fukushima Prefecture 965, Japan

Abstract

This paper presents a novel approach to the reconstruction of geometric models and surfaces from given sets of points using volume splines. It results in the representation of a solid by the inequality $f(x, y, z) \geq 0$. The volume spline is based on use of the Green's function for interpolation of scalar function values of a chosen "carrier" solid. Our algorithm is capable of generating highly concave and branching objects automatically. The particular case where the surface is reconstructed from cross-sections is discussed too. Potential applications of this algorithm are in tomography, image processing, animation and CAD for bodies with complex surfaces.

1. Introduction

There are a number of applied problems that require interpolation or smoothing of large arrays of randomly measured points of a surface. The main sources of such data are physical measurements taken by scanning an object from different viewing directions. Scattered points arise also in mathematical simulation, for example in the solution of the problem of plastic shape forming[1].

This paper presents a novel approach using volume spline technique for interpolating scattered data. It results in a function representation of a solid by an inequality $f(x, y, z) \geq 0$, with the surface defined by the implicit equation $f(x, y, z) = 0$. No preliminary information about topology of the object is needed. Only Cartesian coordinates of points are used in the reconstruction. The particular case where cross-sections of the object are given is discussed too. The volume splines are based on the Green's function and interpolate scalar function values of a chosen "carrier" object.

The main motivation of our approach is to generate a function representation of the reconstructed solids for visualization, inspection and transformation. This representation was shown in[2–4] to be useful for applying various kinds of transformations: set-theoretic operations, offsetting, blending, projection, sweeping, metamorphosis and hypertexturing. The main drawback is that this representation is computation time consuming. We deliberately discuss here parallel implementation and give time performance evaluation of the proposed algorithm.

2. Other Works

We give here a brief overview of several works dealing with related problems, such as scattered volumetric data interpolation, polyhedral and implicit surfaces fitting scattered sets of points, and reconstruction of surfaces from planar contours.

Vasilenko in his monograph[5] considers theoretical issues of spline functions and their practical applications. In particular, a spline function of several variables is described that interpolates scattered data for spaces of various dimensions.

A detailed overview of the techniques for extracting three-dimensional geometries from volume data can be found in[6]. The report contains a comparison of methods of interpolation of scattered volumetric data. The comparison is based upon the test functions and the data sets. For three-dimensional data, domain samples of the data are represented as $\{(P_i, F_i) : i = 1, \ldots, N\}$ where P_i are 3D coordinates, and F_i is a dependent data variable. The compared methods are the modified Shepard's method, volume splines, multiquadrics, the volumetric minimum norm network, and localized volume splines. Global functions methods, such as volume splines and multiquadrics lead to dense systems of equations and are limited to moderate size data sets. The minimum norm network and localized volume splines methods can be applied to processing vast data sets, but do not provide a global analytic description, which causes difficulties in applications.

An interesting approach to the problem of generating a polyhedral surface model from a point-sampled volume data set has been proposed in[7]. The technique starts with a simple model that is already topologically closed, and performs a deformation of the model based on a set of constraints, so that the model grows to fit the object. This is a relaxation process that minimizes a set of constraints. This approach allows the branching problem to be handled without difficulties, but the reconstructed object has to be topologically simple – in fact, homeomorphic to a sphere.

Hoppe et al.[8] describe an algorithm for the reconstruction of a polyhedral surface from an unorganized set of points. The algorithm is based on the idea of determining the zero set of an estimated signed distance function.

Sclaroff and Pentland[9] describe a hybrid implicit/ parametric method based on modal deformations and displacement maps. Sample points have to be one-to-one projected to an initial ellipsoid surface. This imposes restrictions on the shape of the reconstructed implicit surface.

Surface design and computer graphics use mostly parametric curves and surfaces, but lately an increasing attention has been given to geometric design using implicitly defined algebraic curves and surfaces. To fit a surface model to given points, Bajaj et al.[10] solve a constrained minimization problem for some distance criteria. They use an algebraic surface defined implicitly by a single polynomial equation.

Muraki[11] proposes to apply a "blobby" model to fit scattered points. This model describes an isosurface of a scalar field produced by a number of potential field generating primitives. The number and location of the primitives are found by minimization of an "energy function". This method is extremely computationally expensive.

A number of researchers have investigated the problem of reconstruction of surfaces from cross-sectional data[12-14]. The main difficulty in these algorithms is that data are presented as polygonal contours. To connect them into surface polygons is a very difficult combinatorial problem. The surface generated by the triangulation method depends heavily on the choice of the nodes for each contour. The main problem of this approach is the case where different numbers of contours appear in different cross-sections.

A generalized model, called the homotopy model, was presented by Shinagawa and Kunii[15] to reconstruct surfaces from cross-sectional data of objects. Homotopy is used for the reconstruction of parametric surfaces with the help of the toroidal graph representation. The paper[16] presents a homotopy sweep technique. The homotopy is parametrized by two scalar-valued functions to control the transition from one contour to another and the scaling of the cross-sectional shapes. A problem here is that in the homotopy model all points in a contour have their corresponding points in the adjacent contours; therefore, a transformation (or mapping) that defines the correspondence of the points must be fixed.

Let us summarize here the main difficulties that we found mentioned in the papers and encountered in our own experience:

- volume interpolation requires heavy calculations;
- reconstruction of a polyhedral surface representation of topologically complex objects from scanned data involves nontrivial problems of tiling and correspondence of points in adjacent contours;
- description and handling of branching and highly concave structures;
- choice of an appropriate representation for solid objects.

Here we are interested in the representation of complex objects by continuous functions of three variables for subsequent geometric processing rather than for visualization alone.

The next section provides basic information about our approach to reconstruction from scattered points. Section 4 deals with reconstruction from contours. In Section 5, we present experimental results and in Section 6, some discussion.

3. Reconstruction from Scattered Points

Let us start with the formulation of the problems. Let Ω be an n-dimensional domain of arbitrary shape that contains a set of points $\{P_i = (x_1^i, x_2^i, \ldots, x_n^i) : i = 1, 2, \ldots, N\}$. We assume that the points P_i are distinct and lie on or near the surface of an unknown solid. The goal of the reconstruction is to find a smooth function $f(x)$ where $x = (x_1, x_2, \ldots, x_n)$ that approximately describes the solid and its boundary surface.

We represent solid objects by the inequality

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

The function f must have negative values at the points outside the solid, positive values at interior points, and zero values on the boundary surface. Therefore, the surface of the object is defined, in the three-dimensional case, by the implicit equation $f(x,y,z)=0$. This is very convenient for defining simple surfaces like the sphere, the ellipsoid, the torus and others. Two conjunct bodies can be described in terms of set-theoretic union. Analytic representations of set-theoretic operations in the form of so called R-functions have been studied by Rvachev[17] (see also2 and a survey in[18]).

The general idea of our approach is to introduce a *carrier solid* with a defining function f_c and to construct a volume spline U interpolating values of the function f_c in the points P_i. The algebraic difference between U and f_c describes a reconstructed solid. The algorithm consists of two steps. At the *first step,* we introduce a carrier solid object which is an initial approximation of the object being searched. In the simplest case, it can be a ball. Then the carrier function values $\{r_i = f_c(P_i) : i = 1,2,\dots N\}$ are calculated at all given points. At the *second step,* these values are approximated by the volumetric spline derived for random or unorganized points. The problem of finding the interpolation spline function

$$U(P_i) \in W_2^m(\Omega)$$

where W_2^m is the set of all functions the whose squares of all derivatives of order $\le$ m are integrable over $\mathbf{R}^n$, so that

$$U(P_i) = r_i, \qquad i = 1,\dots,N,$$

and

$$\int_\Omega \sum_{|\alpha|=m} \frac{m!}{\alpha!}(D^\alpha u)^2\, d\Omega \to \min$$

where m is a parameter of the variational functional, is considered in detail in5,

The Green's function for the operator T^*T where

$$Tu = \sqrt{\frac{m!}{\alpha!}} D^\alpha u$$

has the form

$$G_{m,n}(x,P_i) = \begin{cases} \|x-P_i\|^{2m-n}\ln\|x-P_i\|, & n \text{ even}, \\ \|x-P_i\|^{2m-n}, & n \text{ odd} \end{cases}$$

where

$$\|x-P_i\| = \left[\sum_{j=1}^{n}(x_j - x_j^i)^2\right]^{1/2}.$$

For 2D and 3D geometric applications (n=2,3 and m=2), the spline has the form

$$U(x) = \sum_{i=1}^{N+k} \lambda_i g_i(x,P_i)$$

where

$$\begin{gathered} g_i(x,P_i) = G_{m,n}(x,P_i), \quad i = 1,\dots,N, \\ g_{N+1}(x,P_i) = 1, \\ g_{N+1+j}(x,P_i) = x_j, \quad j = 1,\dots,k-1, \\ k = (n+m-1)!/n!(m-1)!. \end{gathered}$$

If among the points $P_1,\dots,P_N$ there are k points with the property that for any predefined set of values there is exactly one polynomial of n variables of degree $\le m-1$ that takes these values at these points, then the problem has a unique solution. If m $= n = 2$, then $k = 3$, and it is sufficient to find three points not lying in a straight line.

The minimization of the functional results in the following system of $N+k$ linear algebraic equations (SLAE) for the spline coefficients λ_i:

$$A \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \\ \lambda_{N+1} \\ \lambda_{N+2} \\ \vdots \\ \lambda_{N+k} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where $A = (A_{ij})$ is the matrix with the entries

$$\begin{aligned} A_{ij} &= g_i(P_i,P_j), && i \le N+k,\ j \le N,\ i \ne j \\ A_{ii} &= 0, && i < N, \\ A_{ij} &= g_j(P_i,P_j), && i \le N,\ N < j \le N+k, \\ A_{ij} &= 0, && i > N,\ j > N. \end{aligned}$$

The only problem here is operating with the dense matrix of size N with zero diagonal elements. This matrix is symmetric, but not positive definite. The system may be solved by the Householder method described in[19]. As the coefficients are found, the spline can be restored. The choice of this spline is due to the following reasons:

- it was derived specially for the case of scattered points;
- it provides C^k-continuity if $k < 2m - n$.

Similar splines for 2D case (known as *thin plate splines)* were constructed in[27] without the use of Green's function.

The zero set of the function

$$f(x,y,z) = U(x,y,z) - f_c(x,y,z)$$

for $n = 3$ approximates the unknown surface, and the inequality $f(x, y, z) \geq 0$ represents the unknown solid.

In practical applications the number of points can rise drastically, which limits the applicability of this approach. In the case of a great number of points the spline restoration is very time consuming. One possible way of saving calculation time is the use of parallel computing. A parallel algorithm for volume splines and its performance evaluation were considered in[20]. Results shown in Table 1 demonstrate that the spline approximation problem for a complex surface undergoes parallelization very well on the network of transputers INMOS T800.

4. Reconstruction from Contours

The method of volume splines is limited to sets of data of moderate size because the time of calculation depends on N^3 operations. In many cases, contour points or planar cross-sections are given. The idea of sweeping a 2D closed contour to create a 3D solid object is well known; however, as Hui notes in[21], a major deficiency of the sweep representation is its being restricted by the type of 2D closed contours and the sweep trajectories being used. We do not discuss here the point membership classification algorithms for reconstruction of an object from contours because function representation of a solid solves the problem automatically. The approach proposed here allows us to use two-dimensional contours of arbitrary shapes for reconstructing 3D objects. It helps to decrease the number of random points used for the construction of SLAE and to save computation time. In addition, this algorithm can easily be parallelized. Parallelizing of this algorithm by sharing data between processors can improve computation performance practically in direct proportion to the number of processors.

To reconstruct an object from contours, we apply the above algorithm to each cross-section, using two-dimensional "carriers" and spline functions of two variables $U(x, y)$. Suppose the resultant functions $F_1(x, y)$ and $F_2(x, y)$ represent cross-sections belonging to the planes $z = z_1$ and $z = z_2$ respectively. Intuitively, it is clear that transition and scaling between the cross-sectional shapes can be defined using interpolation or a homotopy map

$$F(x, y, z) = (1 - g(z))F_1(x, y) + g(z)F_2(x, y)$$

where $g(z_1) = 0$ and $g(z_2) = 1$. Note that this method produces a connected surface if the areas $F_1(x, y) \geq 0$ and $F_2(x, y) \geq 0$ have nonempty intersection.

In the case of linear interpolation, $g(z) = (z - z_1)/(z_2 - z_1)$. This well-known method of classical analytical geometry has been applied by Liming[22] in aircraft design. A more complex form, proposed in[16], is a rational polynomial function. To provide a more smooth transition between several cross-sections, one-dimensional spline interpolation along the z-axis can be applied.

Representation of surfaces by sections mainly uses families of parallel slices. On the other hand, some medical imaging systems hardly guarantee parallelism of sections[23]. In this case, we need to interpolate orientational matrices that determine the spatial positions of the sections. Methods for representation of orientational matrices and of smooth interpolation in the set of all orientational matrices are fundamental problems of computer animation, and several methods of interpolation using quaternions are discussed in[24]. In this section we discuss two methods for constructing orientational matrices.

Sometimes the motion of cross sections can be defined in terms of angular velocities. Let γ_1, γ_2 and γ_3 be the direction cosines of the x-axis of the fixed coordinate system in a coordinate system bound with the solid, and $(\omega_1, \omega_2, \omega_3)$ the coordinates of the instantaneous vector of angular velocity $\omega(t)$ in this coordinate system. We assume that we know a sequence of values of $\omega(t)$, and we need to construct a smooth matrix valued function $M(t)$ such that $M(t)$ is the orientational matrix for all t, where t depends on the coordinates (x, y, z) of the solid. If we know the t dependency of (x, y, z), we may solve this problem by numerical integration of the system of differential equations[25]

$$\begin{cases} \dot{\gamma}_1 + \omega_2\gamma_3 - \omega_3\gamma_2 = 0 \\ \dot{\gamma}_2 + \omega_3\gamma_1 - \omega_1\gamma_3 = 0 \\ \dot{\gamma}_3 + \omega_1\gamma_3 - \omega_3\gamma_1 = 0 \end{cases}$$

with the first integral

$$\Gamma = \gamma_1^2 + \gamma_2^2 + \gamma_3^2 = 1$$

and initial conditions

$$\gamma_1{}^0 = \alpha_{11}{}^0, \quad \gamma_2{}^0 = \alpha_{21}{}^0, \quad \gamma_3{}^0 = \alpha_{31}{}^0$$

where $\alpha_{s1}{}^0$ are the entries of the orientational matrix in the initial state. We assume that we have a sequence of orientational matrices in which every matrix is close to the previous one, and we define t by the formula

$$t = \frac{d_1}{d_1 + d_2}$$

where d_1 and d_2 are the distances from the point to the cross-sectional planes.

An orientational matrix has nine entries, and after computation of the entries in the first column we can compute the other six entries.

Let us now discuss the second algorithm for interpolation of a sequence of orientational matrices Mi, $i = 1, \ldots, k$. Assume that we can define t as above. If we use linear interpolation, then the orientational matrix

Number of processors	Time of performance (sec)	Use of processors(%)
1	180.2	99.9
2	90.5	99.4
3	62.5	96
6	32.7	92

Table 1: *The time of solution of the SLAE* (400 x 400)

may be written as

$$M(t) = \exp\big(t \ln(M_i) + (1-t)\ln(M_{i+1})\big).$$

For both interpolation algorithms we assume that the origins of the moving coordinate systems are ordered, say, according to their z-coordinates in the initial frame. The interpolation of orientational matrices allows sections to be calculated through given points by the application of inverse maps.

5. Examples of Data Restoration

We have experimented with reconstruction of objects from data sets obtained from several different sources.

5.1. Fobos Surface Reconstruction

942 points randomly located in the surface of Fobos were used for reconstruction.

The calculations were performed on six transputers T800. The estimation showed that the error of reconstruction of the surface does not exceed 3%. The time characteristics are shown in Table 2. As it is clear from Table 2, solution of the system of linear algebraic equations is the most time-consuming operation. Table 1 exhibits the high rate of parallelization obtained for the problem of solving the system of linear equations.

5.2. Concave Contour Reconstruction

59 points in Figure 2a with Cartesian coordinates *(x,y)* were used to reconstruct an object by our algorithm. The reconstructed 2D "solid" is shown in Figure 2c. Generally, a section may consist of several contours.

Number of points	184	377	628	942
Matrix formation	3.0	11	30	66
Solution of SLAE	6	42	192	633
Calculation of 10000 surface nodes	38	70	113	167

Table 2: *Time characteristics of calculation of the Fobos surface (time in sec.)*

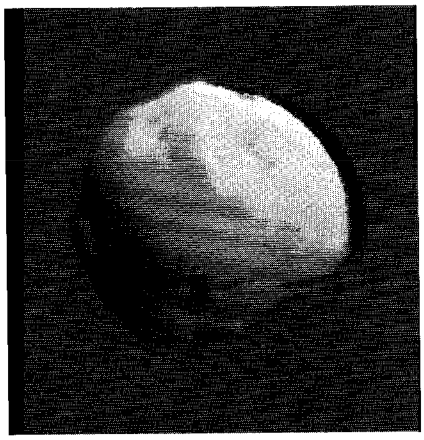


Figure 1: *Fobos surface reconstruction*

Figure 2d shows the result of a section reconstruction from 67 points shown in Figure 2b; here the points form two closed contours.

5.3. Branching Structures

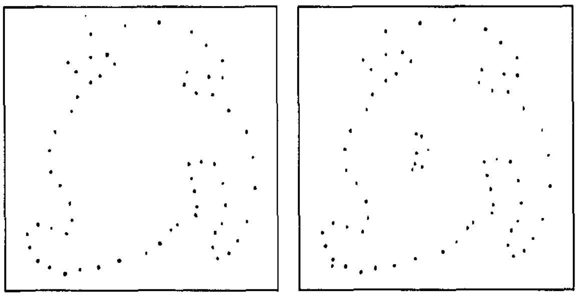
Figure 3 shows a three-branch solid that was reconstructed using the cross-sectional approach.

5.4. Reconstruction of a Solid From Nonparallel Sections

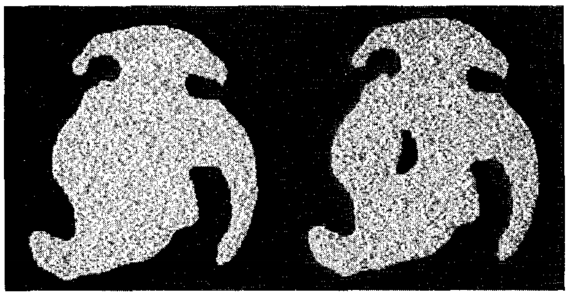
The surface of a bent and twisted block shown in Fig. 4 was reconstructed from 50 nonparallel planar sections, using linear interpolation of orientational matrices as described above.

5.5. Numerical Error Estimate

Numerical error estimation is very important for comparison of various scattered data interpolants because



(a) (b)



(c) (d)

Figure 2: *Concave contour reconstruction*

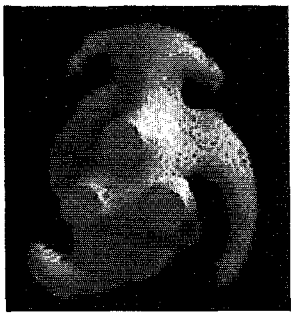


Figure 3: *Branching structure*

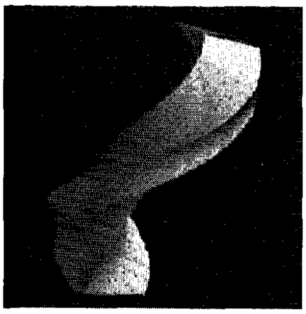


Figure 4: *Example of restoration from nonparallel cross sections*

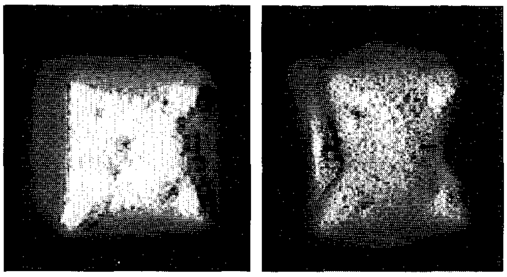


Figure 5: *Reconstructed (a) and deformed cube (b)*

visual evaluation of the result is not sufficient for an appraisal of the algorithm. To compare, some numerical measure of the error of the approximation found for a test function is needed; we used the root mean square measure of error for the test function:

$$F(x,y) = \text{radius}^2 - (x-x_0)^2 - (y-y_0)^2 + \text{noise}(x,y)$$

where

$$\begin{aligned} \text{noise}(x,y) &= s(x,y)s(y,x), \\ s(p,q) &= \text{amplitude} \cdot (s1(p) + s2(p,q)), \\ s1(p) &= \sin(a(p)), \\ a(p) &= \text{frequency} \cdot p, \\ s2(p,q) &= \frac{\text{amplitude}}{1.17} \sin\left(\frac{a(p)}{1.35} + \text{phase} \cdot s1(q)\right), \end{aligned}$$

radius=8.0, $(x_0, y_0) = (5.0, 5.0)$, amplitude=2.0, frequency $= 0.8$, and phase $= 1.0$.

The normalized approximate range of this function over the domain $\{(x,y) : 0 \leq x \leq 10 \quad 0 \leq y \leq 10\}$ is [0.0,8.4]. We used 2500 nodes for restoration of this function. The defining functions of 50 planar vertical slices were combined by using linear interpolation. The RMS error is 0.02 for 469 random test points. The RMS for reconstruction of the same shape with triangular Gregory patches is 0.017[26]. We would like to note that numerical experiments did not show a dependency of accuracy of restoration on the choice of the "carrier" function.

6. Conclusions

We have studied the possibility of use of scattered and cross-sectional data for the reconstruction of solids described by functions of three variables. The new method is based on the spline interpolation of values of a scalar field generated by a "carrier" function. Our algorithm can generate highly concave objects and multiple contours in one slice of data automatically. The algorithm automatically handles branching problems and undergoes parallelization easily. This method can be used for reconstruction and rendering tomographic data. In addition, this technique allows generation of complicated objects that can be easily further modified by application of set-theoretic operations.

We intend to concentrate our future research on the deformation of a carrier solid. Figure 5a shows an example of a cube reconstructed from 90 artificially generated random surface points. In the reconstruction of the cube we used the R-union of two spheres as a carrier solid. Figure 5b shows the deformed cube. The deformation was produced by changing the initial distance between the spheres in the carrier solid keeping the spline function unchanged. We hope to control a plastic or elastic structure like a human face via local changes of a carrier function. Deformations of reconstructed solids by algebraic sums and space mapping using the volume splines introduced as above may be of interest too.

Acknowledgement

We would like to thank the referees for helpful comments.

References

1. V. V. Savchenko, L. G. Sukhomlinov and S. S. Malinowski, "Solving non-linear problems of large plastic planar deformations of sheet metal by using multiprocessor computing", in *Problems of Cybernetics,* Moscow, pp. 67–81 (1982) (in Russian).
2. A. A. Pasko, V. V. Savchenko, V. D. Adzhiev and A. I. Sourin "Multidimensional Geometric Modeling and Visualization Based on the Function Representation of Objects", Technical report 93-1-008 The University of Aizu, Aizu-Wakamatsu, Japan (September 1993)
3. A. A. Pasko and V. V. Savchenko,"Blending operations for the functionally based constructive geometry", in *Proceedings of the CSG 94 Conference, Winchester, U. K. 13-15 April 1994,* pp. 151–161 (1994).
4. A. A. Pasko and V. V. Savchenko, "Solid noise in the constructive solid geometry", in *Compugraphics'93, The Third International Conference on Computational Graphics and Visualization techniques, Alvor (Portugal),* pp. 351–357 (1993).
5. V. A. Vasilenko, "Spline-functions: Theory, Algorithms, Programs", Nauka Publishers, Novosibirsk (1983) (in Russian).
6. A. E. Kaufmann and G. M. Nielson, "Tutorial on volume visualization techniques and applications. Computer Graphics International '93",Ecole Polytechnique Federale de Lausanne, Lausanne (1993).
7. J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara and M. J. Wozny, "Geometrically deformed models: a method for extracting closed geometric models from volume data", *Computer Graphics,* 25(4), pp.217-225 (1991).
8. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuelze, "Surface reconstruction from unorganized points", *Computer Graphics,* 26(2), pp. 71–78 (1992).
9. S. Sclaroff and A. Pentland, "Generalized implicit functions for computer graphics", *Computer Graphics,* 25(4) pp. 247–250, (1991).
10. C. Bajaj, I. Ihm and J. Warren, "Higher order interpolation and least-squares approximation using implicit algebraic surfaces", *ACM Transactions on Graphics,* 11(4), pp. 327–347, (1992).
11. S. Muraki, "Volumetric shape description of range data using 'blobby' model", *Computer Graphics,* 25(4), pp. 227–235, (1991).
12. D. Meyers, S. Skinner and K. Sloan, "Surfaces from contours", *ACM Transactions on Graphics,* 11(3) pp. 228–258 (1992).
13. H. N. Christiansen and T. W. Sederberg, "Conversion of complex contour line definitions into polygonal element mosaic", *Computer Graphics,* 12(2), pp. 187–192, (1978).
14. A. Sunguroff and D. Greenberg, "Computer generated images for medical applications", *Computer Graphics,* 12(2), pp. 196–202, (1978).
15. Y. Shinagawa and T. L. Kunii, "The homotopy model: a generalized model for smooth surface generation from cross sectional data", *The Visual Computer,* 7(2–3), pp. 72–86 (1991).
16. C. L. Tai, K. F. Loe and T. L. Kunii, "Integrated homotopy sweep technique for computer-aided geometric design", in *Visual Computing, T. L. Kunii (Ed.), Springer-Verlag, Tokyo,* pp. 583–595, (1992).
17. V. L. Rvachev, "Methods of Logic and Algebra in Mathematical Physics", Naukova Dumka, Kiev (1974) (in Russian).
18. V. Shapiro, "Real functions for representation of rigid solids", *Computer Aided Geometric design,* 11(2), pp. 153–175 (1994).
19. G. H. Golub and C. E van Loan, "Matrix computation", The John Hopkins University Press, Baltimore (1984).
20. V. V. Savchenko and V. O. Vishnjakov, "The use of 'serialization' approach in the design of parallel programs exemplified by problems in celestial mechan-

ics", in *Performance Evaluation of Parallel Systems, PEPS'93, The University of Warwick, Coventry, U. K.*, pp. 126–133. (1993).

21. K. C. Hui, "Solid modelling with sweep-CSG representation", in *Proc. CSG 94 Conference, Winchester, U. K., 13-15 April 1994*, pp. 153–160 (1994).

22. R. A. Liming, "Practical Analytical Geometry with Applications to Aircraft", Macmillan, New York (1944).

23. Y. L. Kergosien, "Generic sign systems in medical imaging", *Computer Graphics and Applications* 11(5), pp. 46–65, (1991).

24. G. M. Nielson, "Smooth interpolation of orientations", in *Models and Techniques in Computer Animation, N. M. Thalmann and D. Thalmann (Editors), Springer-Verlag, Tokyo*, pp. 75–93, (1993).

25. A. I. Lurie, "Analytical Mechanics", Fizmatgiz, Moscow (1961) (in Russian).

26. K. Miura, "Surface reconstruction with triangular patches", private communication.

27. J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces", in *Constructive Theory of Functions of Several Variables, A.* Dodd *and B. Eckmann (Eds.), Springer-Verlag,* pp. 85–100 (1977).