ISSN 8756-6990, Optoelectronics, Instrumentation and Data Processing, 2018, Vol. 54, No. 4, pp. 334–339. © Allerton Press, Inc., 2018.
Original Russian Text © S.I. Vyatkin, B.S. Dolgovesov, 2018, published in Avtometriya, 2018, Vol. 54, No. 4, pp. 18–25.

ANALYSIS AND SYNTHESIS OF SIGNALS AND IMAGES

Compression of Geometric Data with the Use of Perturbation Functions

S. I. Vyatkin* **and B. S. Dolgovesov**

Institute of Automation and Electrometry, Siberian Branch, Russian Academy of Sciences, pr. Akademika Koptyuga 1, Novosibirsk, 630090 Russia
**E-mail: sivser@mail.ru*

Received March 12, 2018

Abstract—This paper touches upon the problem of a compact polygonal description of objects. A method of lossless compression of geometric data based on perturbation functions is proposed. Advantages of this approach over the known algorithms of transformation of three-dimensional models for fast transmission of information and its compact storage are demonstrated.

Keywords: perturbation functions, theoretic multiple operations, compression of geometric data.

DOI: 10.3103/S8756699018040039

INTRODUCTION

As computer technologies and Internet are developed and the sphere of their applications is extended, there arises an urgent problem of a compact description of complicated three-dimensional (3D) computer models. Databases for such objects can occupy a large volume of memory. Uploading of files (especially from the Internet) can take a long time. Various algorithms of geometric data compression were developed to reduce the memory volume necessary for storage of the polygonal description.

The geometric structure in polygonal models, which is called the geometry, consists of a set of points and topology characterizing interactions between the neighboring points. The description is finalized by the list of attributes (normals, colors, and textures). The majority of geometric data compression methods is based on encoding the relationships of the polygonal presentation of objects and on the description of vertices and faces of the encircling tree by means of vertex reordering [1, 2] or using additional information, e.g., the power of vertices defining the method of vertex addition to the previous sequence [3].

Thus, the list of vertices consists of relations and geometry, where differential encoding or position prediction is used rather than the absolute coordinates of the vertices. The single-rate compression methods of were described in [1–5]. The methods of the so-called progressive geometric compression are extended versions of the single-rate compression methods [6–10]. Naturally, if the models remain within the framework of the polygonal description, high degrees of compression cannot be expected. Thus, a complex VRML file (Virtual Reality Modeling Language) can be reduced by 2.33% from the initial size for the 12-bit quantization and only by 1.67% for the 10-bit quantization.

The description of 3D objects can be made more compact by using special functions. Function-based surfaces are used for many problems in computer graphics, including modeling of soft or organic objects, 3D morphing, detection of collisions, and constructive solid geometry. Though operations for functional objects are simple, form creation is a large problem. Yngve and Turk [11] considered conversion for variational implicit surfaces with the use of an iterative approach. However, transformation of complex models with a resolution of $170 \times 170 \times 373$ voxels required hours of computer operation (R10000, MIPS-processor, 195 MHz, SGI Origin). The iterative approach was also described in [12]. Ohtake et al. [13] presented a method of transforming polygonal models to 3D models on the basis of radial functions. In that method, the polygonal model was approximated by second-order functions, where the sought function was a sum of patches multiplied by the container function (a patch is a quadric surface approximating the surface near the

chosen point). The container function is a factor that guarantees that a patch does not affect other patches outside the chosen domain. The maximum compression ratio in this approach is equal to four.

The goal of the present work is to develop a fast method with a high compression ratio without the loss of geometric data on the basis of perturbation functions.

PERTURBATION FUNCTIONS

Complex geometric objects are usually described with the use of (second-order) functions of deviation from the basic quadric surface [14]. Functionally defined surfaces are constructed from second-order surfaces (quadric surfaces) with analytical perturbation functions, which ensures a high geometric compression ratio for realistic 3D objects. The surfaces are considered as closed subsets of the Euclidean space E^3 defined by the description function $F(x, y, z) \geq 0$, where F is a continuous real function, and x, y, z is a point in E^3, which is defined by the coordinate variables. The function $F(x, y, z) > 0$ describes the points inside the surface, the function $F(x, y, z) = 0$ describes the points on the boundary, and the function $F(x, y, z) < 0$ describes the points that are located outside the domain and do not belong to the surface.

The algebraic inequality of the second degree (with three unknowns x, y, and z) is any inequality

$$F(x, y, z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + A_{13}xz + A_{23}yz + A_{14}x + A_{24}y + A_{34}z + A_{44} \geq 0. \quad (1)$$

This inequality can be written in the matrix form as

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12/2} & A_{13/2} & A_{14/2} \\ A_{12/2} & A_{22} & A_{23/2} & A_{24/2} \\ A_{13/2} & A_{23/2} & A_{33} & A_{34/2} \\ A_{14/2} & A_{24/2} & A_{34/2} & A_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \geq 0, \quad (2)$$

or as a general inequality of the second degree with respect to the spatial variables x, y, and z:

$$(A_{11}x + A_{12}y + A_{13}z + A_{14})x + (A_{21}x + A_{22}y + A_{23}z + A_{24})y +$$

$$+ (A_{31}x + A_{32}y + A_{33}z + A_{34})z + A_{41}x + A_{42}y + A_{43}z + A_{44} \geq 0 \quad (3)$$

($A_{ik} = A_{ki}$, $i, k = 1, 2, 3, 4$).

Free forms are constructed with the use of quadric surfaces and are presented as a composition of the basic quadric and perturbations:

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^{N} f_i R_i(x, y, z). \quad (4)$$

Here f_i is the form factor and $R(x, y, z)$ is the perturbation:

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{if } Q_i(x, y, z) \geq 0, \\ 0, & \text{if } Q_i(x, y, z) < 0 \end{cases} \quad (5)$$

[$Q(x, y, z)$ is the perturbing quadric].

The geometric model generates conditions for designing objects and their compositions of different complexities. For this purpose, we use the set of geometric operations mathematically defined as follows [15]:

$$M^1 + M^2 + \ldots + M^n \to M. \quad (6)$$

Models of complex objects are formed on the basis of perturbation functions with the use of set-theoretic combining and intersecting realized by means of Boolean operations. A binary operation of objects G_1 and G_2 means the operation $G_3 = \Phi_j(G_1, G_2)$ of determining

$$f_3 = \psi(f_1(x, y, z), f_2(x, y, z)) \geq 0, \quad (7)$$

where ψ is a continuous real function of two variables.

Operations of combining and intersecting are of primary interest for solving this problem.

The comprehensive theory of geometric transformations of implicit surfaces was described in [16].

METHOD OF GEOMETRIC DATA COMPRESSION

To ensure a correct transformation and the maximum compact functional description, the initial polygonal model (Fig. 1), which is a combination of several simple geometric models, should be segmented (Fig. 2). To distinguish the basic quadric from the perturbation functions, the coefficients in the equations that describe the model are found in advance. For this purpose, the quadric surface should be inscribed into the simple polygonal model (Fig. 3) or, more exactly, it is necessary to find the coefficients of the basic quadric on the basis of nine points (vertices of the polygonal mesh). The coefficients in the equations of the perturbation functions (1), A_{11}, A_{22}, A_{33}, A_{12}, A_{13}, A_{23}, A_{14}, A_{24}, A_{34}, and A_{44}, are further indicated by the letters $A, B, C, D, E, F, G, H, I$, and K, respectively, and the coefficients of the equations of the basic quadric surfaces are denoted by the letter q. In the next definition of the quadric, we have

$$Q = \begin{pmatrix} q_{xx} & q_{xy}/2 & q_{xz}/2 & q_x/2 \\ q_{xy}/2 & q_{yy} & q_{yz}/2 & q_y/2 \\ q_{xz}/2 & q_{yz}/2 & q_{zz} & q_z/2 \end{pmatrix}. \tag{8}$$

The value of the function defined by Eq. (8) at an arbitrary point $P[x, y, z]$ has the form

$$Q(P[x, y, z]) = q_{xx}x^2 + q_{yy}y^2 + q_{zz}z^2 + q_{xy}xy + q_{xz}xz + q_{yz}yz + q_x x + q_y y + q_z z + q. \tag{9}$$

Fig. 1. Initial polygonal model.

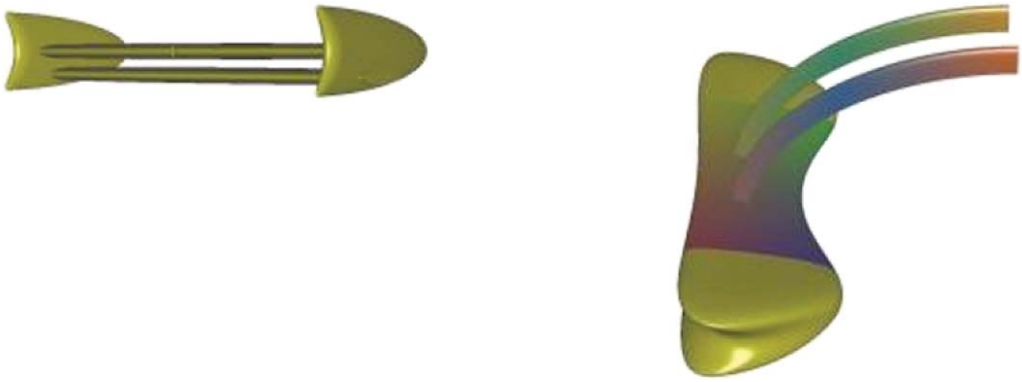

Fig. 2. Simple polygonal objects.

Fig. 3. Simple polygonal (left) and function-based objects.

As the value of Q on the surface is equal to zero, the coefficients of the quadric surface on the basis of nine points ($P_1[x_1, y_1, z_1]$– $P_9[x_9, y_9, z_9]$) are found from the system of linear equations

$$\left\{\begin{array}{c} Q(P_1) = 0 \\ Q(P_2) = 0 \\ \cdots \\ Q(P_i) = 0 \\ \cdots \\ Q(P_9) = 0 \end{array}\right\}. \tag{10}$$

Substituting Eqs. (9) into Eqs. (10), we obtain

$$\left\{\begin{array}{c} q_{xx}x_1^2 + q_{yy}y_1^2 + q_{zz}z_1^2 + q_{xy}x_1y_1 + q_{xz}x_1z_1 + q_{yz}y_1z_1 + q_x x_1 + q_y y_1 + q_z z_1 + q = 0 \\ q_{xx}x_2^2 + q_{yy}y_2^2 + q_{zz}z_2^2 + q_{xy}x_2y_2 + q_{xz}x_2z_2 + q_{yz}y_2z_2 + q_x x_2 + q_y y_2 + q_z z_2 + q = 0 \\ \cdots \\ q_{xx}x_i^2 + q_{yy}y_i^2 + q_{zz}z_i^2 + q_{xy}x_iy_i + q_{xz}x_iz_i + q_{yz}y_iz_i + q_x x_i + q_y y_i + q_z z_i + q = 0 \\ \cdots \\ q_{xx}x_9^2 + q_{yy}y_9^2 + q_{zz}z_9^2 + q_{xy}x_9y_9 + q_{xz}x_9z_9 + q_{yz}y_9z_9 + q_x x_9 + q_y y_9 + q_z z_9 + q = 0 \end{array}\right\}. \tag{11}$$

The coefficients of the quadric surface are calculated by the Cramer method:

$$\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 \\ x_2^2 & y_2^2 & z_2^2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 \\ x_3^2 & y_3^2 & z_3^2 & x_3y_3 & x_3z_3 & y_3z_3 & x_3 & y_3 & z_3 \\ x_4^2 & y_4^2 & z_4^2 & x_4y_4 & x_4z_4 & y_4z_4 & x_4 & y_4 & z_4 \\ x_5^2 & y_5^2 & z_5^2 & x_5y_5 & x_5z_5 & y_5z_5 & x_5 & y_5 & z_5 \\ x_6^2 & y_6^2 & z_6^2 & x_6y_6 & x_6z_6 & y_6z_6 & x_6 & y_6 & z_6 \\ x_7^2 & y_7^2 & z_7^2 & x_7y_7 & x_7z_7 & y_7z_7 & x_7 & y_7 & z_7 \\ x_8^2 & y_8^2 & z_8^2 & x_8y_8 & x_8z_8 & y_8z_8 & x_8 & y_8 & z_8 \\ x_9^2 & y_9^2 & z_9^2 & x_9y_9 & x_9z_9 & y_9z_9 & x_9 & y_9 & z_9 \end{bmatrix} \begin{bmatrix} q_{xx} \\ q_{yy} \\ q_{zz} \\ q_{xy} \\ q_{xz} \\ q_{yz} \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \end{bmatrix}. \tag{12}$$

Solving system (12) with a free term q (which is assumed to be equal to K) for prescribed $P_1[x_1, y_1, z_1] - P_9[x_9, y_9, z_9]$, we find nine sought coefficients: q_{xx}, q_{yy}, q_{zz}, q_{xy}, q_{xz}, q_{yz}, q_x, q_y, and q_z. In what follows, the coefficients of this quadric will be used in the course of transformation to the functional model.

The functional model is defined during rasterization in a cube with the center at (0, 0, 0) and coordinates ranging from -1 to 1 in terms of x, y, and z in the object-fitted coordinate system. Passing around the octal tree

$$\begin{gathered} A' = A/4; \quad B' = B/4; \quad C' = C/4; \quad D' = D/4; \quad E' = E/4; \quad F' = F/4; \\ G' = G/2 + iA/2 + jD/4 + kE/4; \quad H' = H/2 + iD/4 + jB/2 + kF/4; \\ I' = I/2 + iE/4 + jF/4 + kC/2; \quad K' = K/2 + iG/4 + jH/4 + kI/4; \\ K'' = K'/2 + iG'/2 + jH'/2 + kI'/2 \end{gathered} \tag{13}$$

we divide the coefficients A, B, C, D, E, and F by 4, divide the coordinates x, y, and z by 2 (i.e., reduce the cube size), and then apply shifting by the vector $(\pm 0.5; \pm 0.5; \pm 0.5)$, i.e., to one of eight subcubes (13), where the non-primed coefficients are taken from the previous step.

Before transforming polygonal segmented models to functional models, we have to find all points of the polygonal surface. Then all branches and leaves of the tree where the model intersects with subcubes of

Fig. 4. Function-based model.

various levels of the octal tree of object space division are marked. This is easy to do because all intersected leaves of the tree are known:

$$N_x = A'_{14}/\sqrt{A'^2_{14} + A'^2_{24} + A'^2_{34}}, \tag{14}$$

$$N_y = A'_{24}/\sqrt{A'^2_{14} + A'^2_{24} + A'^2_{34}}, \tag{15}$$

$$N_z = A'_{34}/\sqrt{A'^2_{14} + A'^2_{24} + A'^2_{34}}. \tag{16}$$

The essence of the transformation can be described as follows. If recursive division of the object space allows one to visualize functionally defined objects (calculate surface points, normals at these points, luminance, etc.), then one can also solve the inverse problem: find the functions that describe the considered object on the basis of the prescribed points and normals (14)–(16). For this purpose, one has to solve system (13), i.e., calculate the coefficients of the equations of the lowest level of object space division (leaves of the octal tree), pass around the octal tree of object space division in the opposite direction, calculate the function coefficients, and minimize these functions. Systems of linear equations are solved at each recursion level, identical coefficients of the equations are determined (with allowance for the prescribed threshold of the approximation accuracy), and the number of functions at each level is minimized. It is only after all equations of the level are considered that the transition to the next (upper) level occurs. The process is repeated until the root of the octal tree is reached. As a result, a necessary minimum number of functions is obtained, which represent this object in the format of the description of functionally defined objects on the basis of quadric surfaces with analytical perturbation functions. After that, one only has to combine simple objects into a complex object with the use of the set-theoretic operation [15] (Fig. 4).

ANALYSIS OF THE METHOD AND RESULTS

The initial model (see Fig. 1) consisted of 136306 triangles and 68418 vertices; it was converted to 83 functions (see Fig. 4). The description of the surface with $n \times n$ vertices on the basis of polygons requires $3n^2$ real numbers necessary to store these vertices and $6(n-1)^2$ integer numbers necessary to describe the triangles, whereas ten coefficients are sufficient to describe one function in the general case. The considered model consists mainly of ellipsoids, and they are defined by an even smaller number of coefficients. The compression ratio is more than 200. The experimental investigations showed that the compression ratio varied from 10 to 100 and higher depending on the level of detail of the test model; the conversion time varied from several milliseconds to one second. The method was tested for several types of cars, three types of aircraft, many parts, and several assembly units of polygonal models. Using this method, one can transform polygonal models to functional models without losses. Moreover, the image quality will be improved for smooth curvilinear surfaces. If necessary, the model can be converted back to the polygonal description format, as it was done in [17], and then the model can be visualized in a standard manner used for geometric accelerators or in a functionally prescribed form [18]. The tests were performed on the processor Intel Core2 CPU E8400 3.0 GHz. The known approaches to conversion of polygonal models to implicit surfaces involve iterative methods, which are approximate and rather slow. In the proposed approach, the initial polygonal mesh of the complex object is transformed to a function-based model by means of rigorous mathematical calculations without any iterations.

An analyzer of criterial deviations was developed for determining the deviations (vertices and normals). The degree of approximation of the resultant functionally defined surface to the initial triangular mesh was analyzed with the use of two metrics: deviation of the vertices of the triangular mesh from the surface of the functional object and deviation of the normals at the triangular grid vertices from the normals of the functional surface. For this purpose, we first calculated the data of the buffer of the depth of the polygonal

and functional models in the form of a 2D array. Each element of this array is the distance from the camera to the point on the model surface. Then we compared all points of the buffers for finding the mean difference and calculated the mean deviation for the corresponding model. If the polygonal and functional models completely coincide, then this deviation is equal to zero. These results were determined in a 3D cube with a resolution of $1.0 \times 1.0 \times 1.0$, and it was found that compression occurs without losses.

CONCLUSIONS

An effective lossless method of compression of geometric (polygonal) data is proposed. The essence of the method implies the transformation of objects to a functional description on the basis of perturbation functions. The transformation is ensured by means of rigorous mathematical calculations rather than by iterative or other approximate methods, which lead to partial loss of information. The proposed methods of definition of 3D objects and geometric data compression offer a number of advantages over available approaches. The main advantages are the simplicity of the transformation of polygonal objects to the functional description with rapid search for the describing functions and significant reduction of the number of surfaces used to define curvilinear objects.

This work was supported by the Federal Agency for Scientific Organizations (State Registration No. AAAA-A17-117062110016-4).

REFERENCES

1. J. Rossignac and A. Szymczak, "Wrap&zip: Linear Decoding of Planar Triangle Graphs," Computat. Geometry: Theory and Appl. **14** (1–3), 119–135 (1999).
2. J. Rossignac, "Edgebreaker: Connectivity Compression for Triangle Meshes," IEEE Trans. Vis. Comput. Graph. **5** (1), 47–61 (1999). DOI: 10.1109/2945.764870.
3. C. Touma and C. Gotsman, "Triangle Mesh Compression," in *Proc. of the Conf. on Graphics Interface'98*, Vancouver, Canada, June 18–20, 1998, pp. 26–34. DOI: 10.20380/GI1998.04.
4. M. Deering, "Geometry Compression," in *Proc. of the Conf. SIGGRAPH'95*, Los Angeles, USA, August 6–11, 1995, pp. 13–20. DOI: 10.1145/218380.218391.
5. M. Isenburg and J. Snoeyink, "Mesh Collapse Compression," in *Proc. of the 15th Annual Symp. Computational Geometry (SCG'99)*, Miami Beach, USA, June 13–16, 1999, pp. 419–420. DOI: 10.1145/304893.305000.
6. R. Pajarola, J. Rossignac, and A. Szymczak, "Implant Sprays: Compression of Progressive Tetrahedral Mesh Connectivity," in *Proc. of the IEEE Conf. on Visualization'99*, San Francisco, USA, October 24–29, 1999, pp. 299–306.
7. R. Pajarola and J. Rossignac, "Compressed Progressive Meshes," IEEE Trans. Vis. Comput. Graph. **6** (1), 79–93 (2000). DOI: 10.1109/2945.841122.
8. R. Pajarola and J. Rossignac, "Squeeze: Fast and Progressive Decompression of Triangle Meshes," in *Proc. of the Computer Graphics Intern. Conf. (CGI'2000)*, Geneva, Switzerland, June 19–24, 2000, pp. 173–182.
9. E. Derzapf and M. Guthe, "Dependency-Free Parallel Progressive Meshes," Comput. Graph. Forum **31** (8), 2288–2302 (2012).
10. F. Caillaud, V. Vidal, F. Dupont, and G. Lavoué, "Progressive Compression of Arbitrary Textured Meshes," Comput. Graph. Forum **35** (7), 475–484 (2016).
11. G. Yngve and G. Turk, "Robust Creation of Implicit Surfaces from Polygonal Meshes," IEEE Trans. Vis. Comput. Graph. **8** (4), 346–359 (2002).
12. C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and Approximating Implicit Surfaces from Polygon Soup," in *Proc. of the 31th Intern. Conf. ACM SIGGRAPH*, Los Angeles, USA, August 8–12, 2004, pp. 896–904.
13. Y. Ohtake, A. Belyaev, M. Alexa, and H.-P. Seidel, "Multi-Level Partition of Unity Implicits," Proc. ACM Trans. Graph. **22** (3), 463–470 (2003).
14. S. I. Vyatkin, "Complex Surface Modeling Using Perturbation Functions," Avtometriya **43** (3), 40–47 (2007) [Optoelectron., Instrum. Data Process. **43** (3), 226–231 (2007)].
15. S. I. Vyatkin, "Transformations of Functionally Defined Forms," Progr. Sist. Vych. Metody **9** (4), 484–499 (2014).
16. J. Bloomenthal, C. Bajaj, J. Blinn, et al., *Introduction to Implicit Surfaces* (Morgan Kaufmann Publishers, San Francisco, 1997).
17. S. I. Vyatkin, "Polygonization Method for Functionally Defined Objects," Intern. J . Automat., Control Intell. Syst. **1** (1), 1–8 (2015).
18. S. I. Vyatkin, "Recursive Search Method for the Image Elements of Functionally Defined Surfaces," Avtometriya **53** (3), 53–57 (2017) [Optoelectron., Instrum. Data Process. **53** (3), 245–249 (2017)].