Volume XXI 2018
ISSUE no.2
MBNA Publishing House Constanta 2018

Scientific Bulletin of Naval Academy

GPU-based volume-oriented rendering of functionally defined objects

To cite this article: S. I. Vyatkin, Scientific Bulletin of Naval Academy, Vol. XXI 2018, pg. 53-60.

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-18-I2-005

GPU-based volume-oriented rendering of functionally defined objects

S. I. Vyatkin
Institute of Automation and Electrometry SB RAS, Novosibirsk, Russia

Abstract. Volume graphics research involves "re-inventing" computer graphics for volumes and it is applicable to the majority of computer graphics applications. Many applications will be impacted by volume graphics. These include, but not limited to, flight simulations, mission planning, biomedical, environment, industrial and CAD applications. This paper presents a new representation scheme for freeform surfaces it is possible to combine basic surface and perturbation functions. Terrain is represented for the base of scalar perturbation functions.

1. Introduction

Volume graphics is an alternative approach to traditional computer graphics and it is concerned with 3D geometric objects represented as volumes. Unlike conventional computer graphics, which employs continuous surfaces for 3D object representation, in volume graphics a 3D object is represented as a discrete volumetric model, typically stored as a regular volume buffer of voxels. Commonly in volume graphics, the inherently continuous 3D geometric model is discretized (voxelized) in a preprocessing stage generating a view-independent volume buffer of voxels, which then becomes amenable to faster manipulation and rendering using volume graphics manipulation and volume graphics rendering techniques. Unlike surface graphics, volume graphics employs a view-independent model of data and attributes, supports visualization of amorphous phenomena, inner structures, and intermixing of models with sampled and computed datasets, and operations are practically insensitive to object and scene complexities. As such, volume graphics offers a viable and revolutionary alternative to contemporary surface-based computer graphics. Consequently, research in volume graphics might have a substantial impact on the entire field of computer graphics and on its applications. Proposed project would include study and development of enabling methods to represent, model and synthesize, manipulate, and render volume graphics models, develop architecture and system as well an applications of volume graphics.

Research in representation should include studies on the information stored in a voxel as well as the arrangement of voxels in space. In addition to multiple scalar, vector or tensor values at each voxel, attributes and physical properties of the voxel and relationship with neighboring voxels could be stored. Typically, voxels exist in 3D space, but a temporal dimension and additional spatial dimensions may be required for some applications including animation and special effects. Non-regular structured grids, such as curvilinear grids, or even unstructured grids may be more appropriate for representing some objects and should be studied. The tradeoffs between one large volume and multiple volumes to define a scene should be examined. In addition, hierarchical (e.g., recursive multilevel ray casting, "mip-map", and quadrolinear interpolation) and adaptive methods should be studied to efficiently capture fine details, and to allow for multiple resolutions for accelerated manipulation and rendering. The amount of storage required for these structures and manipulation and rendering times can be further reduced with domain technique, such as compression.

In volume graphics, tools that enable the manipulation of rigid and deformable voxel-based bodies are needed. Interactions of interest range from real-time adjustment of visual and material parameters of static objects to physically realistic modeling of collisions between multiple objects or shape changes of non-rigid objects. Manipulation of non-static voxel-based objects includes physics-based modeling of system dynamics, requiring the detection of collisions and calculation of contact and impulse forces, and modeling shape changes of non-rigid objects. Shape changes could involve manipulations that maintain the topology of the objects, such as elastic/plastic deformations and object/image space warping, and manipulations that alter the object topology, such as geometric morphing between objects, and interactive cutting or sculpting of voxel-based objects. A challenging approach is one that avoids re-voxelization.

Studies should focus on efficient local and global illumination techniques that will exploit the volume graphics representation, such as the grid type, and the hierarchical, adaptive, or domain representation. Of excellent interest to volume graphics are special effects, such as penumbra shadows, fuzzy reflections, motion blur, and the like.

2. Flight simulators

Flight simulators require efficient processing large databases describing modeled environment in real time. Algorithms that solve correctly tasks in this application area can be defined as algorithms of visualization of open database. Term *open database* means that whole database does not fit in view field. New and promising arm systems make principally new requirements to computer image generators (CIGs) used in simulators for environment synthesis. Most complicated tasks are low height flights following terrain, officer staff training, helicopter pilot training etc. Design of new CIGs require solution of such problems as:

- Improving realism of synthesized images due to increasing system performance and due to implementing new visual effects (texture of different kinds, haze, smoke, rain, snow, 3d clouds etc.)
- Imaging large terrain areas (400x400 km and more).
- Projecting images on screens with large view field angle (up to 360 degrees) with moving viewer and projecting device. Projecting images on spherical screen of large size is challenging and complex task. Its solution is required primarily for army simulator applications. Complexity and volume of computations increase dramatically for moving viewer and projection device.
- Imaging terrain areas using elevation grid. Systems solving this task by polygonal triangulation are known. Computational power of modern visualization systems is not more than 15000 polygons per frame in one channel and for realistic synthesis of mountain landscape; it is required at least ten times more.
- Animation and deformation objects and surfaces. Possibility to model such natural phenomena as waves water fountains and terrain surface movement due to bombing, ships and submarines going under water. Change of clouds in the sky etc.

Solutions of these problems by traditional approaches in systems produced by the industry leader result in increasing system complexity and pricing.

3. Freeform objects

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 E^n with the definition:

$$f(x_1, x_2, \ldots, x_n) \geq 0, \tag{2}$$

where f is a real continuous function defined on E^n.

A functionally defined object is completely defined by means of the real-valued describing function of three variables (x_1, x_2, x_3) in the form of $F(X) \geq 0$, then the objects are treated as closed subsets of the Euclidean space E^n, defined by the describing function $F(X) \geq 0$, where F is the continuous real-valued function and $X = (x_1, x_2, x_3)$ is the point in E^n, 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 [1].

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

where x, y and z are spatial variables.

The freeform is a composition of the base surface and the perturbation functions

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

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

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

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

Since $\max[Q + R] \leq \max[Q] + \max[R]$, 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 (Figure 1).

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 $F(X) \geq 0$, we can visualize not only the surface but also the internal structure of the object.

4. Non-polygonal terrain

Unlike the cartoonish appearance of terrain in most polygon-based simulators, volume-oriented flight simulation can handle in real-time high level of detail called for when training weapon system officers and pilots in flight operations. Generating accurate updated worldwide databases on a daily or even hourly basis encompassing 3D view of the mission theater is increasingly critical to mission planners and tacticians. Planning mission logistics, navigation paths and strategies depends on extreme accuracy in terrain elevation, image reconstruction and interaction. Volume graphics provides a framework for producing images of nearly photographic quality - the equivalent of 10's of millions of textured polygons.

Figure 1. Terrain, B-2 Spirit and scattered light

Terrain visualization is a difficult problem for applications requiring accurate images of large datasets at high frame rates. Known non-polygonal methods of photorealistic relief visualization are quite slow. Attempts to increase speed by different types of acceleration methods (hierarchical [2], parametric [3], a massively parallel computer [4] or special parallel ray-casting hardware [5], hybrid ray-casting and projection technique [6] improve the situation, but still do not achieve real-time speed for high performance terrain visualization.

In order to render voxel-based terrain, proposed method must be able to convert a 3D scalar field representing the terrain into a set of vertices and triangles that can be rendered by the graphics hardware. A method for constructing a triangle mesh whose vertices coincide with the zero-valued isosurface is the Marching Cubes algorithm [7]. Although it provides many greater capabilities, the use of voxel-based terrain in real-time virtual simulations also introduces several new difficulties. The algorithms used to extract the terrain surface from a voxel map produce far greater numbers of vertices and triangles when compared to conventional 2D terrain. The development of a seamless LOD algorithm for voxel-based terrain is vastly more complex than the analogous problem for height-based terrain. Texturing and shading of voxel-based terrain is more difficult than it is for height-based terrain. In the cases that triangle meshes are generated for multiple resolutions, arises the cracking problem. A method for patching cracks on the boundary plane between cells triangulated at different voxel resolutions was described in [8].

Using a voxel-based model [1], however, can achieve the same results at a much lower hardware requirement. As a software solution, the method is portable so it can be integrated into any flight simulation system regardless of hardware architecture.

This paper describes results of some investigations concerned with modeling of a mountainous landscape in which it is proposed to use voxel-based terrain without triangulation.

The open simply connected set of points on the plane a domain of the plane was introduced in [1]. Let D be the plane domain and $\overline{\mathrm{D}}$ its closure. Let’s enter the coordinate system (u, v) on the plane. Let x, y, z be the rectangular Cartesian coordinates of the points in the 3D Euclidean space E^3. Prescribe three continuous functions on the set $\overline{D}$:

$$x = \varphi(u,v),\ y = \psi(u,v),\ z = \chi(u,v), \tag{6}$$

Further, assume that functions (6) have the following property. If (u_1, v_1) and (u_2, v_2) are different points of the set $\overline{D}$, then $M_1(x_1, y_1, z_1)$ and $M_2(x_2, y_2, z_2)$ of the space E^3, whose coordinates were calculated by formulas (3), are also different:

$$\begin{aligned} &x1 = \varphi(u1, v1),\quad y1 = \psi(u1, v1),\quad z1 = \chi(u1, v1), \\ &x2 = \varphi(u2, v2),\ y2 = \psi(u2, v2),\ z2 = \chi(u2, v2), \end{aligned} \qquad (7)$$

The set S of the points $M\ (x,y,z)$ whose coordinates x, y, z are defined by (6), where the functions φ, ψ, χ in the closure $\overline{D}$ of the domain D possess the described property, is called a simple surface. The simple surface that is a plot of the function defined in the 3D space $z = f(x, y)$ is referred to as the freeform surface F. The terrain representation based on the scalar field is a totality of a base surface P (in the same coordinate system as F) and the related altitude map. Any surface may use as the base surface, however, surface used in practice are simple surfaces such as planes, ellipsoids, or cylinders. The altitude map is a 2D rectangle called hereafter a perturbation domain D_P of the base surface P, and the perturbation function $h(u,v)$ is given inside this rectangle. The altitude map in turn determines the perturbation. The domain of $h(u,v)$ is $D_{h(u,v)} = \{U, V\}$, where U and V are the size of the rectangle. The altitude map and the base surface are related as follows: there exists a transformation $G(\Re^3 \Rightarrow \Re^2)$ from the coordinate system of F and P to coordinate system of the map. This transformation is usually a parallel projection. The value of $h(G(d_F))$ characterizes the deviation of the point d_F, on the surface F from the point dp that is the projection of this point onto the surface P. In other words, the value of $h\ (G(d_F))$ is equal to the scalar of vector

$$\vec{v} = (\vec{d}_F - \vec{d}_P) \qquad (8)$$

Therefore, the domain of the terrain can be defined as a set of point in $\Re^3$, which are defined by the vector equation

$$\vec{F} = G(\vec{v}) + \vec{n}\cdot h(G(\vec{v})); \forall \vec{v} \in \Re^3, \qquad (9)$$

where $\vec{n}$ is the normal to the base surface.

If the vector $\vec{v}$ is outside the perturbation domain, the vector $\vec{n}\cdot h(G(\vec{v}))=0$ and $\vec{F}$ is the vector on the base surface. Thus, for prescribing the form of the perturbing surface, we can use a table of numbers, and the function h can be represented by a function of interpolation by pivotal values taken from the table. In this case, we may assume that a scalar field is given in the perturbation domain D_p. The function h has the form:

$$h(u,v) = f_0 + (f_1 - f_0)(v - m\_v), \qquad (10)$$

where

$$f_0 = (1-(u-m\_u))\text{table}[m\_u, m\_v\ ]) + (u-m\_u)\text{table}[m\_u+1][m\_v],$$

$$f_1 = (1-(u-m\_u))\text{table}[m\_u, m\_v+1]) + (u-m\_u)\text{table}[m\_u+1][m\_v+1],$$

where m_u is the integer part of u, m_v is the integer part of v, and table$[m_u][m_v]$ is the m_uth and m_vth elements of the table.

The terrain F based on the scalar field is specified by means of the surface and the perturbation function (the table of numbers, which characterizes the deviation of the surface F from the base one at checkpoints). This paper considers representation of volcano Bandai based on the base planes. In this case, the transformation G is a parallel projection directed oppositely to the normal vector of the base plane. We will use the notion of the terrain F as a combination of the base planes and the perturbation domain; it may have a rectangular contour or be defined by vector equation (9).

Terrain is defined by means of the basic plane and the perturbation function defined in an infinitely long parallelepiped. Values of the perturbation function are specified at the parallelepiped cross-section by a 2-D height map. As a basic surface, we may use a plane, and then the direction of the carrier plane normal must match the longitudinal direction of the parallelepiped - the region of perturbation function definition.

Since during rendering it is necessary to estimate the maximum function on a three-dimensional or one-dimensional interval, then maps of the level of detail are preliminary composed for efficient calculation. The initial data form the level n if the array dimension is 2^n x 2^n. Data for the level n-1 are obtained by choosing a maximum from four adjacent values of the level n, the rest three values are not considered further, i.e., we obtain a 2^{n-1} x 2^{n-1}array. The zero level consists of only one value, that is, the maximum all over the height map.

5. Volume traversal

The rasterization process is divided into two stages and distributed between the central processor and the graphics processing unit. The central processor divides the object space according to a quaternary tree.

In the search algorithm, the cube is divided into smaller parts, for which the object intersection test is performed. The division process takes two steps. The first step is when the cube is divided into four parts in the plane *XY*.

Then each part is considered separately. If there is no intersection with a given object, then this part is neglected further on. If there is intersection with other parts, the same division procedure is repeated. In the general case, the process is finished when the part under consideration corresponds to a tile of a certain size. The advantage of this approach is that, at an early stage, one can neglect the larger parts of the cube that do not have a given object. The primitives of the intermediate description are the fragments where geometric objects intersect with tiles.

The second step of computations includes processing the list of objects and determining the visibility and pixel color, both of which are carried out by the graphics processing unit. The fragments of the object are fed to the input of the graphics processing unit. Then the fragment is tested for intersection with a ray directed along the *Z* axis, and the binary search for the nearest intersection point of the ray with the object is carried out [9]. The problem is to find the first point at which the function vanishes. Having determined such a point for each ray, one can determine the *z* coordinate. Then the normal is determined in each pixel.

Having all coordinates and normals in each pixel, one can apply a local lighting model. The result is an image of a smooth object with account for lighting.

The visualization time is reduced by the effective use of the computational resources of the NVIDIA graphics processing unit with a CUDA architecture. The method was implemented with account for the influence of the speed of operation with memory. Registers are used to a maximum degree, and memory is used jointly. In all other cases, the processors work with the general memory of the graphics processing unit.

The functions of the graphics processing unit included calculating the coordinates of the surface points, normals, and lighting. The central processor carried out the geometric transformations, rasterized primitives in the tile grid, and formed a list of fragments with determination of all the necessary parameters. For the visualization, the DirectX applied programming interface was used. The testing was carried out using the Intel Core2 CPU E8400 3.0 GHz and GPU 470 GTX processors.

6. Conclusions

Conducted research of volume-oriented visualization technology gives rise to creation of a new kind of visualization systems for different application areas.

In the proposed methods, the following problems have been solved:

- Terrain skinning;

- Distortion correction (DDC);
    - NLIM, FOV near 360, aperture;
    - Non-Static viewpoint position and attitude;
    - Non-static projector position and attitude;
- Light points;
- Landing lights;
- 3D morphing;
- Atmospheric effects (fog, haze, rain, snow, layered atmospheric effects, round earth, sky illumination)
- Phong model in object space;
- Shadows;
- Levels of detail (LOD);
- Interpenetrating;
- Conforming;
- Collision detection;
- Transparency;
- Animation;
- Deformation;
- Moving objects.

Additional features of methods:

- Feature texture;
- 3D texture;
- Shape texture;
- Procedural texture;
- Thematic texture;
- Volumes;
- Quadrics;
- Freeform surfaces;
- Arbitrary order of object processing;
- Subpixel filtering with aperture;
- Multilevel masking;
- Multilevel ray casting.

Proposed visualization methods have a number of advantages as compared with existing systems, namely:

- Number of surfaces for rendering non-planar objects is greatly decreased. Defining volumes by freeform surfaces allows one to compress data base size 100 and more times as compared with polygonal representation.
- Distortion correction is greatly facilitated. Let us note that there is a majority of non-planar surfaces in visualization system implementing distortion correction, since after distortion of planar surfaces on screen they become non-planar. Yet non-planar surfaces can hardly become planar ones after distortion on the screen.
- Geometry processor (CPU) load is decreased greatly as well as data flow from geometry to video processor (GPU).
- Voxel-based rendering solves almost all problems pertaining to photorealistic imaging of terrain by elevation grid without preliminary triangulation.
- Time of landscape processing and rendering does not depend much of elevation map resolution.
- Simplicity of surface animation and deformation.

- Possibility to visualize volumes.
- Wide range of possible application areas.

Technique of volume visualization has advantages due to spatial ordering in viewer space – fast ray stop, advantages due to ordering in object space – regular volume traversal. This technique elegantly combines possibility of freeform surfaces and heterogeneous volumetric areas.

Algorithmic solutions proposed in this project to process three-dimensional data intended for real time visualization comprise basis for creation new generation of computer image generator. Project of discussed system also uses the best properties of well-known techniques, such as virtual visualization, procedural texturing (thematic texturing). Obtained results make it possible to create system for different application areas with high level of image accuracy and high processing speed. Due to adaptiveness of the technique to processing volumes distributed in space more efficient use of visualization for scientific experiments, medical research etc. becomes possible.

References

[1] S.I. Vyatkin, Complex Surface Modeling Using Perturbation Functions, *Optoelectronics, Instrumentation and Data Processing,* Volume 43, Number 3, 2007, pp. 226-231.

[2] D. Cohen and A. Shaked, "Photo-realistic imaging of digital terrain," in *Proceedings of Eurographics 93, Barselona-Spain, 6-10 September 1993,* n.d, pp. 363-373.

[3] D. W. Paglieroni and S. M. Petersen, "Parametric heights field ray-tracing," in *Proceedings of Graphics Interface,* 1992, pp. 192-200.

[4] G. Vezina and P. K. Robertson, "Terrain perspectives on a massively parallel SIMD computer," in *Proceedings of CG International 91, Springer-Verlag*, 1991, pp. 163-188.

[5] P. Pitot, Y. Duthen, and R. Caubet, "A parallel architecture for ray-casting," *Computer Graphics,* vol. 89, pp. 463-472, 1989.

[6] G. Agranov and C. Gotsman, "Algorithms for rendering realistic terrain image sequences and their parallel implementation," in *Proceedings of Graphicon*, 1995, pp. 153-161.

[7] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Computer Graphics, Proceedings of SIGGRAPH 87,* n.d, pp. 163–169.

[8] R. Shu, C. Zhou, and M. S. Kankanhalli, "Adaptive marching cubes," *The Visual Computer,* vol. 11, pp. 202–217.

[9] S. I. Vyatkin, "Method of Binary Search for Image Elements of Functionally Defined Objects Using Graphics Processing Units," Avtometriya 50 (6), 89–96 (2014) Optoelectron., Instrum., Data Process. 50 (6), 606–612 (2014)].