Voxel-Based Terrain Generation Using Scalar Perturbation Functions

Sergei I. Vyatkin, Boris. S. Dolgovesov
Institute of Automation and Electrometry, SB RAS, 630090
1 prosp. Koptiuga, Novosibirsk, Russia
sivser@mail.ru,
bsd@iae.nsk.su,

ABSTRACT

The problem of real-time photorealistic imaging of digital terrain is discussed. New technique for specifying digital terrain by scalar perturbation functions without its approximation by polygons is considered. A recursive algorithm for object space subdivision with masking of invisible surfaces and an effective technique of projective transformation for perspective imaging are proposed. The possibility to visualize photo-realistic terrain is shown. Examples of images obtained by modeling the work of the algorithm with elevation map of defining the digital terrain are presented.

KEYWORDS: Scalar Perturbation Functions, Height Maps, Voxel-Based Terrain

1. INTRODUCTION

Terrain visualization is a difficult problem for applications requiring accurate images of large datasets at high frame rates, such as flight simulation. Rendering photo-realistic, complex terrain features at interactive rates requires innovative techniques. A polygonal model and geometric pipeline can be used but this introduces massive storage requirements and, ideally, a parallel implementation of the algorithm. However, features with high spatial frequency context (ridge lines and canyons) require large numbers of polygons to meet a specified level of terrain accuracy. Using a voxel-based model, 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. Image generators traditionally use polygons as database primitives. It is difficult to use polygons to make convincing models of clouds, smoke, trees, and anything else for which no simple surface representation exists. The analysis of possible directions of evolution of a real-time visualization systems shows that the easiest way to improve picture quality, i.e. to increase number of polygons rendered per frame, is not the most effective one. Going this way, the qualitative changes can be hardly achieved. For instance, introduction of texture was such a change. Even displaying much more polygons, an image without textures will be more poor than an image with less polygons but with color textures. Using traditional polygonal representation for the example complex surface give rise to a range of problems such as visible surface determination, depth complexity handling, controlling levels of details, clipping polygons by viewing frustum, geometry transformations of large number of polygons. Known methods of photorealistic relief visualization are quite slow. Attempts to increase speed by different types of acceleration methods (hierarchical [1], parametric [2], a massively parallel computer [3] or special parallel ray-casting hardware [4], hybrid ray-casting and projection technique [5]) improve the situation, but still do not achieve real-time speed for high performance terrain visualization. Thus, we have first devised an algorithm that is fast enough on an ASIC which can be easily parallelized [6]. In this paper we present results of some investigations concerned with modeling of a system in which it is proposed to use voxel-based terrain. The possibility of high performance terrain visualization is investigated. Terrain is represented for the base of scalar perturbation functions.

2. VOXEL-BASED TERRAIN VISUALIZATION

It is proposed to describe complex geometric objects by defining (in the scalar form) the second-order function of deviation from the basic surface or (in the simplest form) from the basic plane [7]. A terrain is a particular case of such objects; it 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

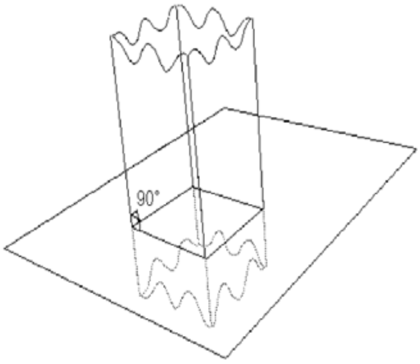

Figure 1. The region of perturbation function definition

map. As a basic surface we may use a plane, then the direction of the carrier plane normal must match the longitudinal direction of the parallelepiped - the region of perturbation function definition (See Figure 1).

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. The process of preparing the maps of levels of detail is shown schematically in Figure 2.

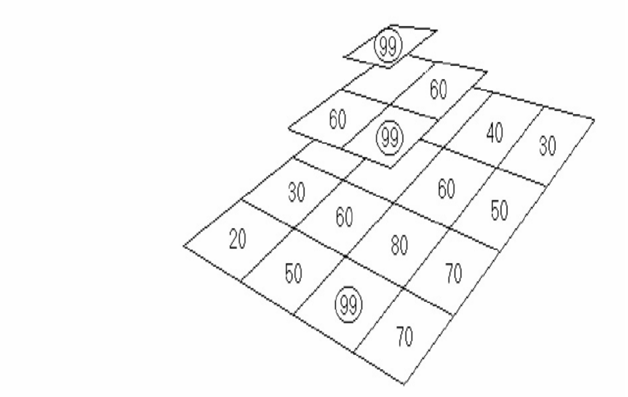

Figure 2. Maps of levels of detail

While determining the perturbation maximum, we calculate the characteristic size of the current interval projection, which governs the level of detail. A cruder approximation of the initial function is chosen for a larger interval. If a more accurate representation is required, then we perform bilinear or bicubic interpolation of values of heights from the last level of detail. Therefore a terrain model is coded as differential height map, i.e. the carrier surface is defined by algebraic means and only deviation from this basic surface is stored in the each node. Such a modeling method simplifies creation of smooth detail levels and shading. The data of height grid is not subject to geometry transformations as the triangle vertices do. The geometry transformations are only required for the carrier surface. During the recursive voxel subdivision on each level, we project the centers of the voxels onto basic plane.

The computed coordinates, as well as in the case of ordinary RGB texture map, will define address in the so called "altitude map" or "shape texture" [8]. We calculate the altitude corresponding to this address and a level of details, and use it to modify coefficients of the plane or quadric equation. As a result we will obtain a smooth surface of arbitrary shape modulated with the values from the altitude map. But the problems solved by this algorithm requires much more complicated methods within the traditional approach. Indeed, the common way to present terrain with polygons requires an abundance of polygons. Besides, the number of additional problems arises such as high depth complexity, hidden polygons removal, priorities, switching between levels of detail, clipping polygons by the pyramid of vision, etc. Such problems do not appear in the proposed method. It is proposed to describe terrain by defining the function of deviation from the basic plane. Terrain is constructed on the basis of plane. Terrain is a composition of the basic plane and the perturbation functions $F'(x,y,z) = F(x,y,z) + R(x,y,z)$, where $R(x,y,z)$ is the scalar perturbation function. Put another way, R is a number computable by projection given voxel on the height map. First, we use quaternary subdivision terrain (see figure 3), as it is described in [9]. On the last quaternary tree level find coordinates of univariate bar - voxel V_0, which will be assigned pair vectors $P_0=(x0,y0,z0)$ and $P_1=(x1,y1,z1)$, $V_0=\{P_0,P_1\}$.

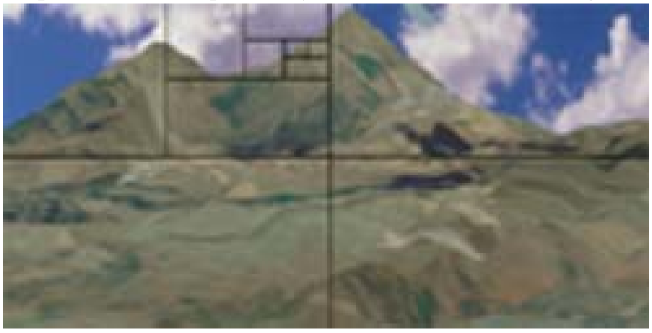

Figure 3. Quad Tree Subdivision

Further, coordinates of voxel V_0 by means of transformations G are converted in coordinate system height map:

$$\{(x0,y0,z0),(x1,y1,z1)\} \Rightarrow \{(u0,v0,h0),(u1,v1,h1)\}.$$

We use matrix transformation T in the height map coordinate system, which being multiplied to the matrix of geometric transformation M and gives a resulting matrix of transformation G. G=T*M;

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, voxel transformed coordinates (u, v, h, a) in coordinate system of height map are calculated from (x,y,z) voxel coordinates in model space by multiplying a vector of point in model space to matrix G.

$$G^* \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ h \\ a \end{bmatrix}$$

Further, we use voxel subdivision on Z coordinate or binary voxel subdivision. At this stage, for the current level of recursion end vector of voxel being nearest with respect to the observer, is supposed equal nearest end vector of voxel preceding level subdivision. Far-away vector of voxel is calculated as a semi-sum of vector amount of near and far-away voxel preceding level subdivision.

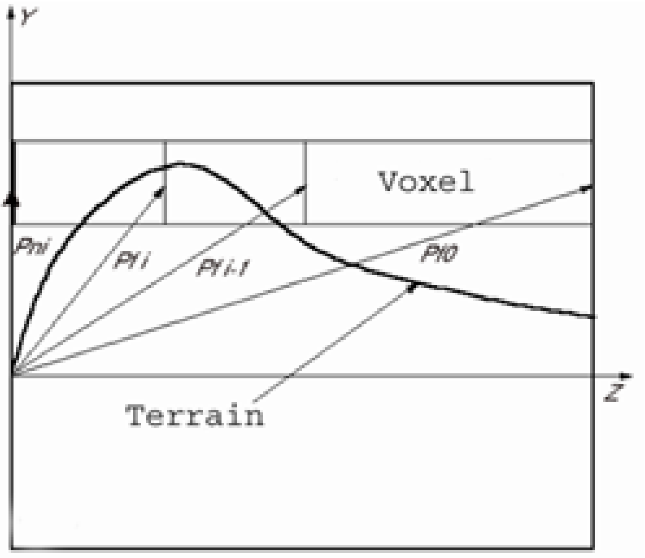

Figure 4. Binary Subdivision

$$P_{n\,i} = P_{n\,i-1}\,,\ P_{f\,i} = (P_{n\,i-1} + P_{f\,i-1})/2\ .V_i = \{P_{n\,i},\ P_{f\,i}\},$$

where V_i is a voxel of i –level of recursion, $P_{n\,i}\ P_{f\,i}$ is the coordinates of near and far-away voxel of i-level subdivision.

Given process is illustrated on Figure 4. By sizes of voxel projections corresponding recursion level is calculated level of detail. By u and v coordinates of points $P_{n\,i}$ and $P_{f\,i}$ is realized sample of maximum value from table presenting given level of detail.

Calculated therefore number is a value of perturbation function of base plane.

On each stage voxel subdivision on its sizes is calculated level of detail. If level of detail not last level, then calculated height h is compared to value of height of given level Hmax, and if h>Hmax, then voxel subdivision stops.

1. define size of rectangle being voxel projection on the height map as a maximum of distance from the point {u0,v0} to the point {u1,v1}-Lp (See Figure 5);

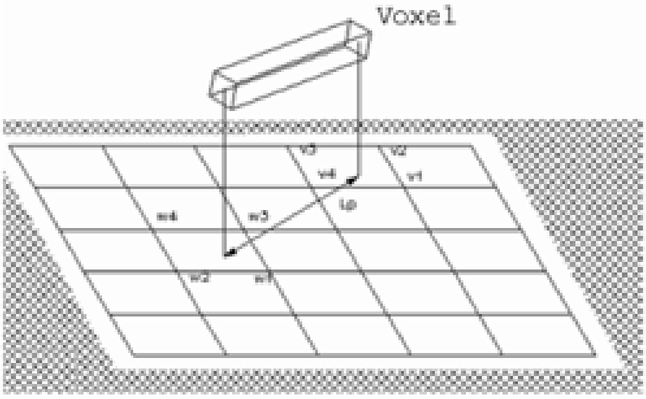

Figure 5. Voxel projection on the height map

2. from the inequality

   $\frac{1}{2^{level}}$ <Lp< $\frac{1}{2^{level+1}}$ define a level of detail ('level').

Figure 6 shows a result of voxel-based terrain modeling without preliminary triangulation with bilinear interpolation of height values (height map resolution – 200x200).

3. SUMMARY

Our investigation in the volume-oriented visualization technology have made it possible to reveal some advantages in both the scene representation technique and the rendering algorithm oriented to real-time implementation. The main merits of our approach are the following:

- reduction of the load on the geometry processor and decrease of data flow from it to the video processor;
- the geometry processor works with the single basic plane;
- the right priority order is provided by the corresponding traversing of the tree and the set of masks;
- sufficiently simpler construction of terrain because the preliminary surface triangulation and the viewing pyramid clipping are unnecessary (to change the level of detail we use a mechanism similar to the usual texture sampling);
- the computation time in terrain generation is practically independent of the height map resolution and depends only on the screen resolution (See Table, Figure 7 and Figure 8);
- simple animation and deformation of terrain.

All these merits of our approach form the ground for creating a new class of computer visualization systems for various applications. Preliminary estimates have shown that for implementation of the systems it is desirable to develop a custom VLSI (ASIC) [6].

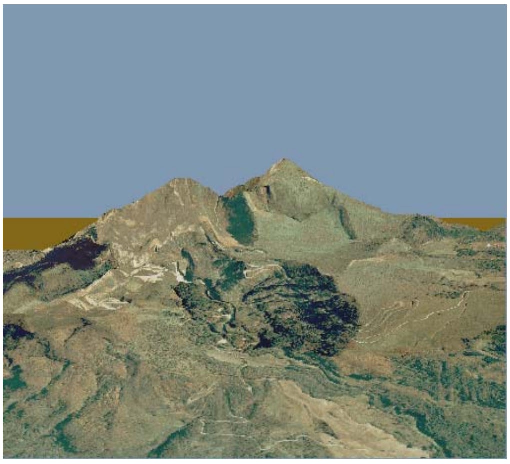

Figure 6. Voxel-Based Terrain

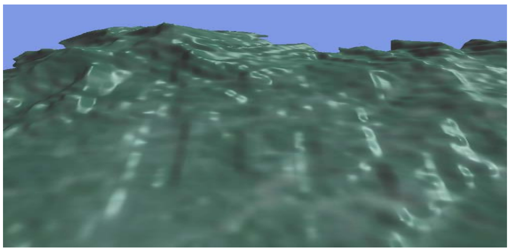

Figure 7. Height map with resolution 64x64

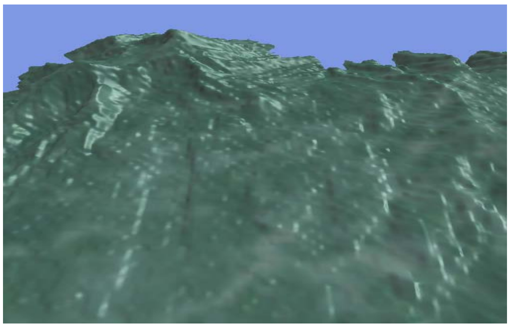

Figure 8. Height map with resolution 128x128

Table:

```
Intel Pentium Celeron 333 Mhz
settings QLevels x BLevels: 9x21 (512x512
pixels 21 z depth)

height map resolution| render time sec |
picture
1024x1024            | 59              |
t1024.tga
512x512              | 58              |
t512.tga
128x128               | 58              |
t128.tga
64x64                | 57              |
t64.tga
```

REFERENCES

[1] D. Cohen and A. Shaked "Photo-realistic imaging of digital terrain". *Proceedings of Eurographics'93*, Barselona-Spain, 6-10 September 1993, Volume 12 Number 3, pages 363-373.

[2] D. W. Paglieroni and S. M. Petersen "Parametric heights field ray-tracing". *Proceedings of Graphics Interface'92,* 1992, pages 192-200.

[3] G. Vezina and P. K. Robertson "Terrain perspectives on a massively parallel SIMD Computer". *Proceedings of CG International'91*, Springer-Verlag, 1991, pages 163-188.

[4] P. Pitot, Y. Duthen, and R. Caubet "A parallel architecture for ray-casting". *Computer Graphics'89*, pages Blenheim, 1989.

[5] G. Agranov and C. Gotsman "Algorithms for Rendering Realistic Terrain Image Sequences and Their Parallel Implementation". *Proceedings of Graphicon'95*, pages 153-161, St. Petersburg, 1995.

[6] Vyatkin S., Dolgovesov B., Yesin A., et al., Parallel architecture and algorithms for real-time synthesis of high-quality images using Voxel-Based surfaces // Graphicon' 2000: *10th International Conference on Computer Graphics,* Moscow, August 28- September 2, 2000, pages 117-123.

[7] Vyatkin S., Dolgovesov B. Surfaces and Patches of Arbitrary Forms Based on Scalar and Analytical Perturbation Functions // Graphicon' 2002: *12th International Conference on Computer Graphics,* Nizhny Novgorod, 2002.

[8] Vyatkin S., Dolgovesov B., Ovechkin V., et al., Photorealistic imaging of digital terrains, freeforms and thematic textures in realtime visualization system Voxel-Volumes, *GraphiCon '97*, Moscow. 1997, pages 121-126.

[9] Vyatkin S., Dolgovesov B., Yesin A., et al., Voxel Volumes volume-oriented visualization system, *International Conference on Shape Modeling and Applications* (March 1-4, 1999, Aizu-Wakamatsu, Japan) IEEE Computer Society, Los Alamitos, California, 1999, pages 234-241.

About the author

Sergei I. Vyatkin (Ph.D.) is a scientific researcher of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry, SB RAS.

His contact email is sivser@mail.ru,

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry, SB RAS.

His contact email is bsd@iae.nsk.su.