Terrain Generation using Fractals

Pedro Costa
University of Minho
Department of Informatics

Categories and Subject Descriptors:

-

1. INTRODUCTION

For the past thirty years, terrain generation has been a deeply researched field due to the wide variety of applications from military training to animation, with a special obvious impact in game development. The demand for new, unknown spaces in virtual environments is constant, and increasingly more exigent in realism. Manual modelling is not a viable option in these cases, as the processes, tools and techniques used are laborious, repetitious and often require speciallized 3D modelling skills.

Similar work to the one done for this document has already been done recently by [Smelik et al. 2009], with a high focus in modern trends. This document is primarily based on that document but adds more recent work ([Chen and Shi 2011]), and digs deeper into the origins of this computer graphics branch. Although this field has been in development for more than three decades, the earlier works ([Fournier et al. 1982], [Miller 1986]) are still heavily used and impose a requirement to understand newer approaches.

2. PROCEDURAL STOCHASTIC MODELLING

The usage of procedural modelling to generate virtual terrains has been an active research topic in computer graphics for more than thirty years. Instead of designing content by hand, procedural modelling allows the design of a procedure to automatically create the content. The applications are various and not restricted to the automatic generation of virtual spaces (such applications are out of the scope of this document though).

This field of research has been majorly focused on the random generation of terrains, either from scratch or constrained by the user. It is obvious that the randomness associated with this generation process can not be achieved with deterministic processes alone. Also due to their visual randomness, realistic natural objects can not be obtained with deterministic modelling. The number of samples required to achieve a visually acceptable model would easily create prohibitively large model databases, and the level of detail available would always be strictly binded to its size.

In their work, [Fournier et al. 1982] defined the formal concept of a stochastic model as a model where the object is represented by a sample path of some stochastic process of one or more variables. These models have three requirements:

(1) an appropriate object (or phenomenon) to be modeled;

(2) a stochastic process to model it with;

(3) an algorithm to compute the sample path of the process;

In the same document, the authors also identify a terrain as the example of an origin for a stochastic model. The processes associated with the models can be analitically defined. Therefore, many traditional algorithmic techniques, such as recursive subdivision, may be used to compute the required sample paths.

The recursive subdivision technique was introduced in [Catmull and Clark 1978] for parametric patches, and the advantages are even more obvious in stochastic modelling, for two reasons:

—It allows an infinite detail (one can always get more detail from the model), but at the same time, no more than the necessary detail is generated for a given scale;

—The basis of the computation is an interpolation formula, which is, in general, much easier to compute than an incremental one, especially for midpoint interpolation.

2.1 Primitives

Two properties required for a modelling primitive to be useful:

Internal consistency : the primitive must be reproducible in any position of an appropriate coordinate system at any level of detail, i.e., its features must not depend on its position and/or orientation in space;

External consistency : adjacent modelling primitives must provide continuity if a common boundary is to be shared.

Both these properties are intuitively easy to preserve in deterministic modelling. Yet, in a stochastic modelling primitive these are more difficult to maintain and require more care in the design of generating algorithms.

3. FRACTALS

Natural 3-D terrain generally presents two features:

Infinite detail : the amount of detail captured by the vision system is the only limitation, as closer points of view will reveal smaller details;

Self-similarity : as one closes in, the details revealed in the terrain are similar to those shown at distance (the creases of a rock seen when very close it remind the ridges of mountains).

These features are precisely those of fractal geometry. The generation of terrains based on fractals is a deeply studied subject, and is the basis of most modern approaches.

In the generation of a terrain, the presence of infinite detail can be obtained with the already mentioned recursive subdivision methods. At great distance, the amount of detail necessary is very small, resulting in a lower number of required polygons. When the distance decreases, new polygons can be generated in the required area by subdividing the polygons.

As for the self-similarity, [Mandelbrot and Van Ness 1968] introduced the term "fractional Brownian Motion" (fBm) to denote a family of one-dimensional Gaussian stochastic processes which provide useful models for many natural time series. A sample of these functions may be observed at any scale, always showing the same statistical features. Thus, a surface generated using fBm would possess macroscopic features up to the order of magnitude of the overall surface generated, as well as arbitrarily small surface detail.

In [Fournier et al. 1982], the authors used the fBm as the basis for their recursive subdivision algorithm. This algorithm starts from a limited domain and continuously subdivides it until a resolution threshold is reached. The image of a given point in the subdivided domain is then defined as the interpolation of the adjacent points (the first step uses the domain limits) displaced by a Gaussian function with zero mean and unit variance, with a constant seed. The influence of the generated displacement is decreased in each iteration. The decrease factor depends on a variable h which controls the roughness of the result. To note that this algorithm has linear complexity, proportional to the number of points to be generated, with a rather small number of operations in each computation.

The application of Fournier's algorithm in two-dimensional primitives gives origin to three distinct algorithms: two based in polygon subdivision (for triangles and for quadrilaterals, respectively) and one for a stochastic parametric surface. For both the polygon subdivision cases, some care must be taken to ensure required properties of the modelling primitives. Internal consistency implies that the random number generated is independent of both the position and current scale. This requires the seeds of the random number generator to be indexed by some invariant identifier of the points rather than by any function dependent on their positions, and the random numbers to be generated in the same order at each level of the subdivision.

External consistency, on the other hand, implies the displacements for the points generated on a boundary shared by two polygons to be equal. This can be accomplished again by tying the seeds of the random number generator to the invariant point identifiers (making sure that the same identifiers are assigned to the corresponding points in the representation of each polygon's boundary). However, if the displacements are allowed to follow the polygon's normal, non coplanar polygons will cause discontinuities. The solution is to define a point's normal as the average of the normals of all the polygons containing it. Since this problem persists even for generated points which are completely internal to an original polygon, one can just assume the normal of the original polygon as the direction for the displacements.

3.1 Algorithms

The first algorithm presented in [Fournier et al. 1982] is meant for scenes in which all surfaces consist of triangles, and is known as the **triangular edge subdivision**. Each triangle can be subdivided into four smaller triangles by connecting the midpoints of each side, creating a pattern similar to a Sierpiński Sieve fractal (see Figure 1).

Fig. 1. The Sierpiński Sieve fractal.
Source: `http://mathworld.wolfram.com/SierpinskiSieve.html`

With the same philosophy, the algorithm for quadrilaterals generates the midpoint of each side. Opposed midpoints are then connected, and the midpoints for both connections are generated. These two points are also connected, and the midpoint of this connection is taken to be the central point of the quadrilateral subdivision. The pattern created is also similar to a known fractal – Sierpiński Carpet – which can be seen in Figure 2.

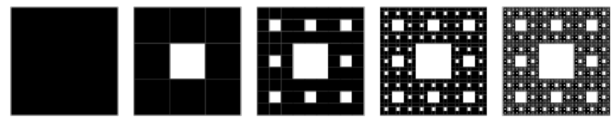

Fig. 2. The Sierpiński Carpet fractal.
Source: `http://mathworld.wolfram.com/SierpinskiCarpet.html`

The stochastic parametric technique proposed by [Fournier et al. 1982], also known as the **diamond-square subdivision**, is the basis of current algorithms, and is still used for the generation of terrain meshes with no constrains. The algorithm starts with the four corners of a quadrilateral terrain (usually a square) and works in two steps. First, in the diamond step, it generates the central point of the quadrilateral. This point is assumed to join four diamonds (in the first step, only half of each diamond is visible). Second, in the square step, the central point of each diamond is generated. All the points now form a grid of squares, and the process may be repeated until the required resolution is reached.

Gavin Miller, in [Miller 1986], describes the diamond-square algorithm as flawed. In his website[1], Robert Pendleton explains that Miller's complains were due to the fact that he wanted to constrain the central point of the terrain – if he left this point to be generated along with all the others, "even he would've had to admit that the algorithm works pretty decently as a terrain generator".

In his work, Miller provides an alternative to Fournier's algorithms. His alternative, which he calls **square-square subdivision** starts with a square base. The first step generates the points of a square with half the area of the original, with the same center. For each point, the value is taken to be in the proportion 9:3:3:1, where the nearer points have a greater weight in the interpolation. For each original square, the internal generated points are then connected to those of adjacent squares, resulting in a continuous. The first impression is that this algorithm wastes the border points of the original squares, but these will be useful in the next iteration, regaining use of the whole area.

Figure 3, Figure 4 and Figure 5 contain the diagrams for the explained algorithms, taken from [Macklem 2003].

[1] `http://gameprogrammer.com/fractal.html`

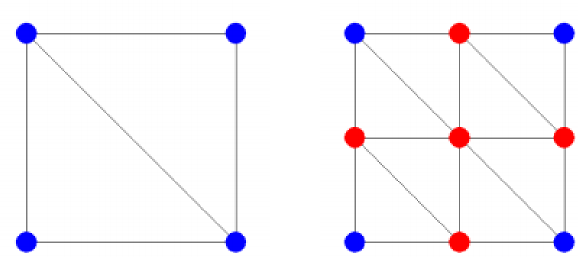

Fig. 3. One step of the triangle-edge subdivision. In blue, the original points. In red, the generated ones.

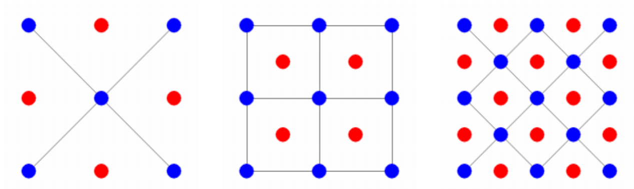

Fig. 4. One full iteration of the diamond-square subdivision. For each step, the blue points are the originals and the red ones are the new generated points.

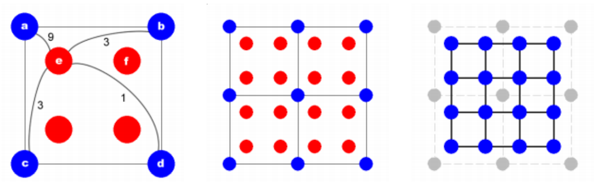

Fig. 5. One step of the square-square subdivision. In blue, the original points. In red, the generated ones.

4. HEIGHT-MAPS

4.1 Methods

5. WATER

6. VEGETATION

7. ROADS

8. URBAN ENVIRONMENTS

9. CONCLUSION

REFERENCES

CATMULL, E. AND CLARK, J. 1978. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10,* 6, 350 – 355.

CHEN, J. AND SHI, X. 2011. Study of real sense terrain generating based on fractal. *Procedia Engineering 12*, 54–59.

FOURNIER, A., FUSSELL, D., AND CARPENTER, L. 1982. Computer rendering of stochastic models. *Commun. ACM 25*, 371–384.

MACKLEM, G. 2003. Computer landscape generation and smoothing.

MANDELBROT, B. B. AND VAN NESS, J. W. 1968. Fractional brownian motion, fractional noises and applications. *SIAM Review 10,* 4, 422–437.

MILLER, G. S. P. 1986. The definition and rendering of terrain maps. *SIGGRAPH Comput. Graph. 20*, 39–48.

SMELIK, R., DE KRAKER, K., TUTENEL, T., BIDARRA, R., AND GROENEWEGEN, S. 2009. A survey of procedural methods for terrain modelling.