Real-time Rendering of Fractal Rocks

Wolfgang Deix*

Institute of Computer Graphics
Vienna University of Technology
Vienna / Austria

Abstract

This paper introduces a possible fractal model for rocks and stones where the model's emphasis is laid on fast and random generation. The concept is based on the similarity of rock surfaces and terrain surfaces which brought forth the idea of creating rocks with well-researched landscape and terrain generation algorithms.

Keywords: Computer graphics, modeling of natural phenomena, artificial rocks, rock models, fractal Brownian motion, midpoint-displacement

1 Introduction

Intensive research of realistic looking landscape creation with Fractal Brownian motion (FBm) goes back to 1980 when Loren C. Carpenter introduced a *fast* and *random* terrain generation method based on midpoint-displacement (first introduced by Archimedes (287 - 212 BC) for parabola construction). The idea of rendering computer generated landscapes (in real-time) began to evolve. [1].

Since then in this, in terms of computer science, long period of time many additions for terrain and landscape generation have been made that had all originated from one basic problem: Images of landscapes cannot be reproduced with simple visualization of just the landscape's surfaces. The Need for Detail arose.

For instance in 1988 Benoit B. Mandelbrot pointed out that past fractal forgeries of landscape fail to include river networks [2] which inspired P. Prusinkiewicz and M. Hammel to build "A Fractal Model of Mountains with Rivers" [3].

Summaries of often requested features of synthetic landscape images include skies, clouds, plants, trees, architectural buildings, weather influences (e.g. snow), stones and rocks. Surface textures are commonly used to take those into account and "to enhance visual detail in the image beyond what can be modeled geometrically" [4].

With increasing performance of graphics hardware, algorithms that render complex models out of predefined simple ones and hardware technology utilizing these algorithms, we are now in the delicate situation that we'd like to have at least simplified models of those (usually surface textured and not geometrically present) details. (Algorithm example: surface subdivision, hardware technology example: Truform™ by ATI Technologies Inc.).

In the case of artificial rocks we can exploit the similarity between terrain surfaces and rock surfaces. Well researched terrain generation algorithms can be used to create models of rocks. These rocks can be inserted into an artificial landscape (e.g. spatial-distributed by another FBm) to enhance its entire visual impression.

2 Midpoint-Displacement

The midpoint-displacement method used in this paper is completely adopted from the midpoint-displacement with triangles algorithm revisited in [3], as shown in *Figure 1*. To surpass an upcoming problem arising from the fact that rocks and stones are not completely self-similar (see also *Self-Similarity*), it is important that we handle the context-sensitive nature of the midpoint-displacement method with the deterministic variant of this technique, suggested by Smith [5], illustrated in *Figure 2*. When an edge is subdivided, the coordinates of its endpoints determine, via a hashing function, an index into a pre-stored table of random numbers that represent possible displacement values. So if the displacement value of the midpoint of edge `l` in *Figure 3* is calculated separately for both triangles `P` and `Q` that share `l`, the returned values y_P und y_Q will be the same.

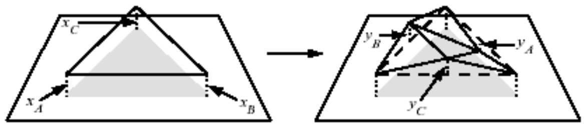


Figure 1: Midpoint-displacement method with triangles ([3]).

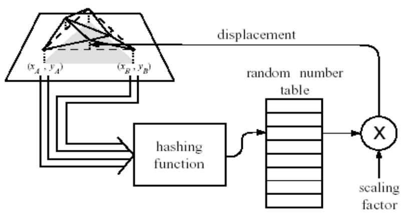


Figure 2: Deterministic calculation of a pseudorandom midpoint-displacement value ([3]).

* wolfgang.deix@gmx.at

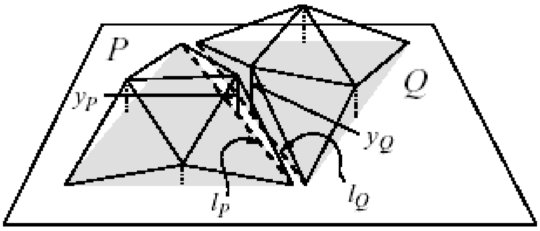


Figure 3: The midpoint displacement method is context-sensitive ([3]).

3 Subdivision

The basic subdivision algorithm is used to approximate complex surfaces with a predefined accuracy (e.g. approximating curved surfaces) [6]. The former mentioned midpoint-displacement method for triangles can be seen as an enhanced subdivision method. In the case of creating smooth and rounded artificial rocks we can use the rough displaced midpoints as control points of interpolated or approximated splines that represent each curved surface. These curved surfaces will be approximated again by subdivision as a refinement step. So we have at least 2 (interpolation) or 3 (approximation) subdivision steps:

1. midpoint-displacement subdivision to create control points.
2. (only approximation): new midpoints that lie on approximation spline.
3. new subdivision points between neighboring mid- and endpoints and mid- and midpoints.

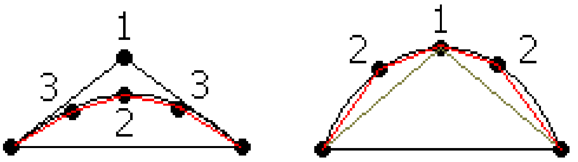


Figure 4: Generated subdivisions points for approximated (left) and interpolated (right) curves.

4 Self-Similarity

Rocks usually have a very limited degree of self similarity, the scaling property of fractals is not applicable. The usual spherical-like closed rounded shape of rocks is not self similar. In many cases upsides and downsides of rocks can be identified which is again a not self similar feature. Therefor we have to find solutions for the modeling of the not self similar parts of artificial rocks.

In this approach we consider a pre-modeling step that creates the basic, not self similar shape of our artificial rocks: We choose a sphere as basis and use an icosahedron, shown in *Figure 5*, to approximate it with arbitrary detail (subdivision can be easily applied on icosahedra). On this basis we apply the fractal modelling algorithm which creates the random surfaces. To differentiate between upsides and downsides we will use different roughness constants *H* (so called Hurt constant [7]) for faces pointing up and faces pointing down in our midpoint displacement algorithm (The Hurt constant *H* determines the possible range of the displacement of our midpoints. H is between $0 < H < 1$, displacement curve smoothenss for *H* going towards 1. For further explanation consult [8]).

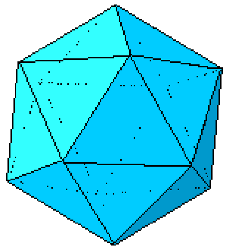


Figure 5: Icosahedron, platonic solid

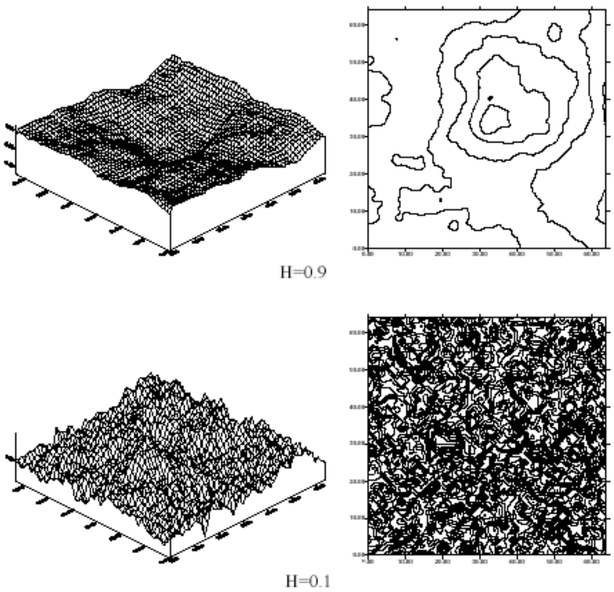


Figure 6: Constant H = 0.9 (upper) and constant H = 0.1 (lower).

The bottom surface of artificial rocks will more likely get a smooth shape, the upside can have a more coarse surface. We determine the value of constant H for each face due to the correlation of its normal vector and the perpendicular normal vector of the XZ plane in object space (vector in Y axis direction in object space), as shown in figure 7.

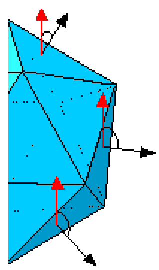


Figure 7: Correlation angles between face normals and XZ plane normal.

5 Texture Mapping

Möller et al [9] define texture mapping as "a process that takes a surface and modifies its appearance at each location using some image, function, or other dataset". For instance if we want do create a wooden wall, instead of creating each wood-part of the wall seperetly we take a simple rectangle or quad and paint just an image of these wooden parts (texture) on it. This is the simplest case where an image is wrapped on polygons and each texel (compare pixels of a texture) modifies directly the color component of the corresponding vertex inside the polygon. Usually a function is used to obtain texture coordinates to corresponding vertices (e.g. planar, spherical, cylindrical, cubic projection).

Special additions and enhancements such as multitexturing allow more sophisticated effects to be realized: specular lighting look-alike with shininess textures, reflections with environment mapping or uneven surfaces with bump mapping.

6 Bump Mapping

Bump mapping is a technique that makes a surface appear uneven in different definable kind: bumpy, wavy and many others [9]. In contradiction to texture mapping where only color components are changed, bump mapping perturbs the surface normal. Modifying the normal has an impact on the shading calculation and thereby changes the perception of the polygon surface itself. This allows simulating small bumps on a surface.

There are two basic methods for defining the normal variation [9]:

- Bump map defines 2 signed values at each point that modify the normal along u and v axes.
- A height field is defined at each point and is used to derive the u and v values for modifying the normal. These are calculated by taking the differences between neighboring values.

7 The Recipe - Putting the Stones together

We initialize our object model with a simple icosahedron structure with e.g. 20 faces (=triangles), set a certain randseed for the current rock that we will create and fill our hash table for random displacement values with randomly generated numbers.

We want to render our artificial rocks in real-time. That is why it is not possible to create models with infinite detail. We will use bump mapping to add the details that we cannot generate geometrically. For this we have got two possibilities: (1) use a static bump map to create just a fake rocksurface illusion, (2) dynamically pre-calculate a bump map for each individual rock. In the second case we need additional preparation steps so nothing is missing when actually rendering (see dynamic bump map preparation below).

Rock model preparation

We process one random midpoint displacement iteration for triangles on every face of the icosahedron. On every existing edge we interpolate a spline curve through endpoint – (new) midpoint – endpoint. We subdivide this curve so that we get 5 resulting vertices (endpoint – first subdivision – midpoint – second subdivision – endpoint). We create 3 new edges between every midpoint and again create midpoints on these new edges (which I will call mid-midpoints for distinction). Now we can create 16 triangles with following patterns, see also *Figure 8*:

- 1 endpoint and 2 neighboring subdivision points. (3x)
- 2 neighboring subdivions points and 1 mid-midpoint. (3x)
- 1 subdivision point, 1 midpoint and 1 mid-midpoint. (6x)
- 1 midpoint and 2 mid-midpoints. (3x)
- 3 mid-midpoints. (1x)

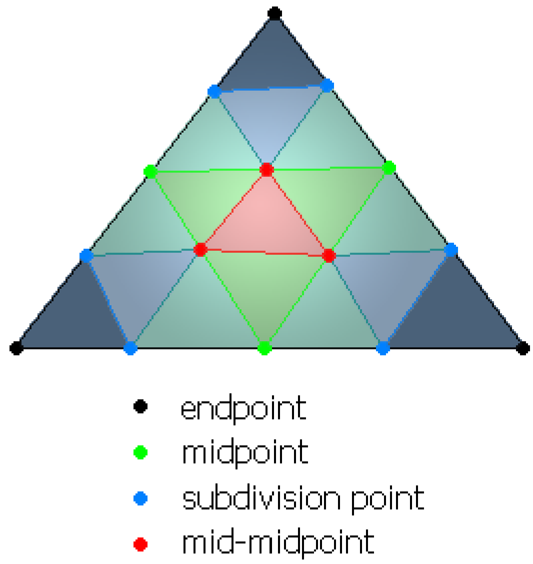


Figure 8: Creating 16 sub-triangles after midpoint-displacement and subdivision damping.

Dynamic bump map preparation

This will take a certain setup-up time for each rock. We execute our midpoint-displacement and subdivision damping to a more detailed stage than we do for our model which is used for real-time rendering. So we iterate the above mentioned rock model preparation more times to get a more complex object model (e.g. 2 – 3 times will be sufficient). From this created geometric model we calculate a unique normal map, bump map that we will use later on the much more simplified geometric model of this rock. For this we have to average normals on every vertex (from normals of neighboring faces) and use the texture coordinate look-up function to write the resulting normal into our normal map. This step is only valid if the simple and the complex model generation process use the same hash table of random displacement values so that both models correspond to each other.

Rendering: Using a multitexturing approach, with an image texture that creates the basic grainy stone look combined with our normal map used for bump mapping, will give the finishing touch.

8 Performance considerations

When starting with an icosahedron that consists of 20 triangles, we'll get 320 triangles after the first midpoint-displacement & subdivision damping iteration. After the second we'll end up with already 5120 triangles per artificial rock which is probably too much if we want to produce a detailed scene that consist of around 100 artificial rocks additionally. Texture mapping and bump mapping will help us to keep the triangles per artificial rock ratio low. The number of midpoint displacement & subdivision damping iterations in fact has to be choosen arbitrarly due to the proportion of actual size and amount of rocks used in our scenery. For hundreds of small rocks one iteration should be enough.

In the test environment the *subdivision-damping* process scaled better than the *midpoint-displacement* process.

Process	Triangles	Duration	Vertex/ms
Midpoint	20	64,25 ms	0,9338
Midpoint	320	391,25 ms	2,4536
Midpoint	5120	4477,75 ms	3,4302
Subdivision	20	41 ms	1,4634
Subdivision	320	146,25 ms	6,5641
Subdivision	5120	632,25 ms	24,2941
Dyn.Bump.	20	209 ms	0,287
Dyn.Bump.	320	581 ms	1,6523
Dyn.Bump.	5120	4589 ms	3,3471

Performance table for different work processes.

9 Further research

There are interesting fields of research that would fit together perfectly with artificial rocks.

Displacement Mapping is a method to present surface detail by defining an offset (displacement) from a base surface [10]. It differs from bump mapping in actually modifying the surface geometry instead of perturbing the normals and can also be done hardware accelerated. ***Solid Texturing*** is a method to look-up a texel for a vertex in a complete texture volume [12]. For instance with real-time procedural solid texturing [13] a technique can be used that utilizes hardware accelaration for real-time rendering. Solid texturing would be very suitable to simulate broken rocks or rocks with crevices.

Cave Creation: Building artificial rocks can also be used as a basis for cave creation. Randomly created and randomly concatenated rocks can define the border walls of caves. If we substract these concatenated rocks out of a quadric volume, a random cave can be visualized.

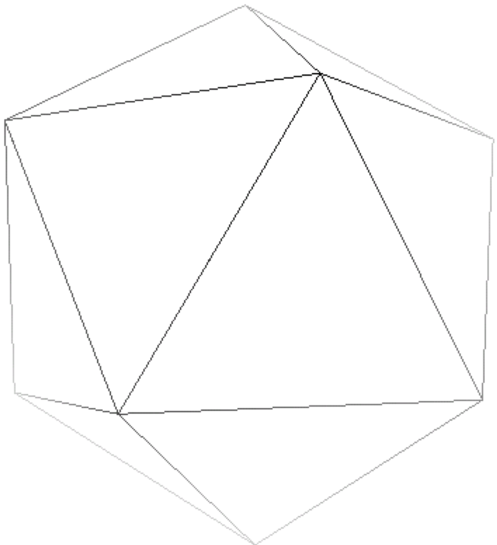
Conclusion

To simulate realistic looking landscapes specific detais are needed to achieve convincing results. Rocks being one small part of these details are usually completely realized in painted textures of landscape surfaces. With increasing graphics hardware power it will be possible to display bigger rocks and stones even with simplified geometric models.

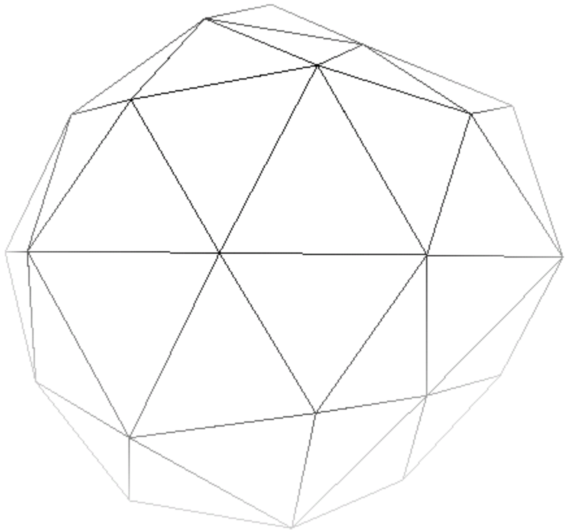
Starting with a spherical basic shape and applying a well known terrain generation algorithm like midpoint-displacement, with an orientation adaptive roughness constant, and the enhancement of a special smoothing step like subdivision damping, is one way how geometric rock approximation models can be created. Simple models used for real-time rendering can be visualy improved with texture and bump mapping.

The visual quality can be additionally improved with more challenging techniques as displacement mapping and solid texturing. Artificial rocks can be used as basis for more sophisticated building algorithms like cave creation.

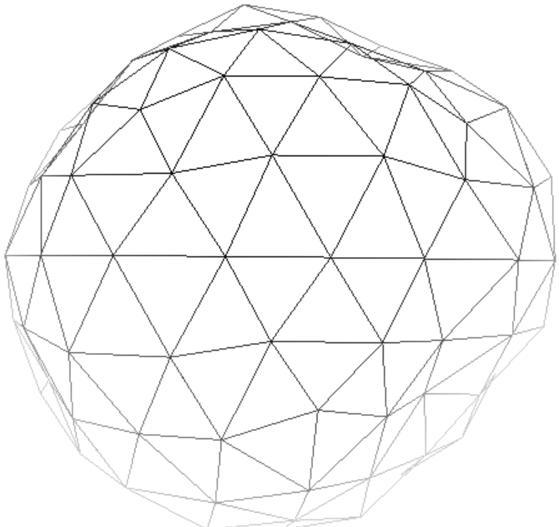
Images



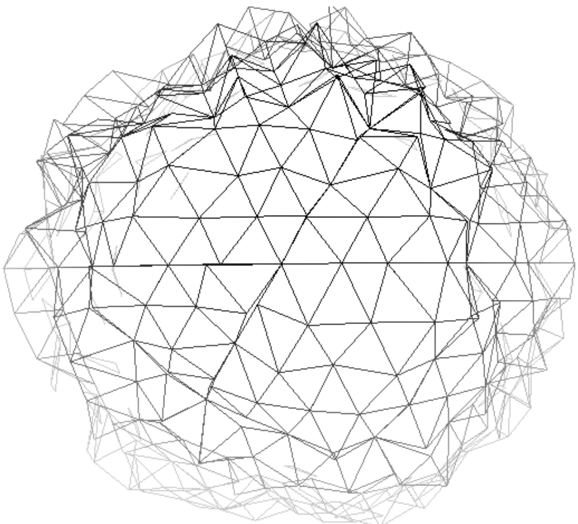
Icosahedron as basis model



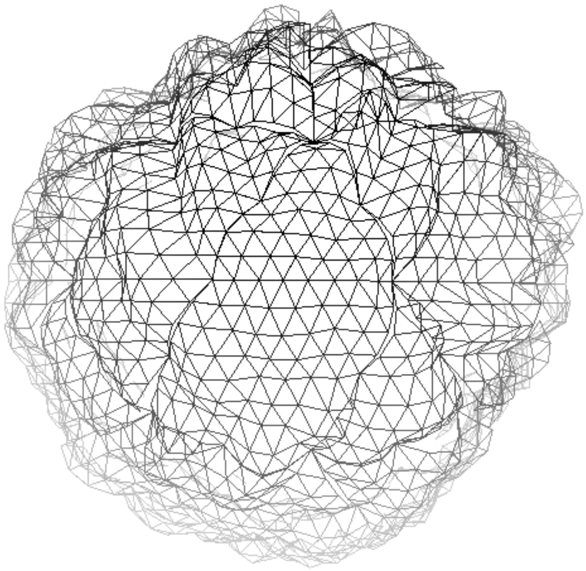
First midpoint-displacement step



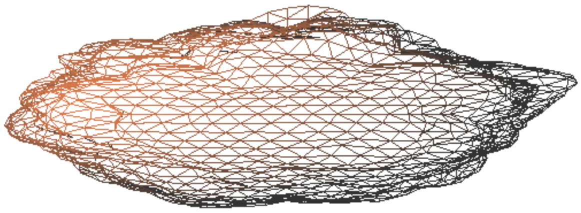
First subdivision-damping step



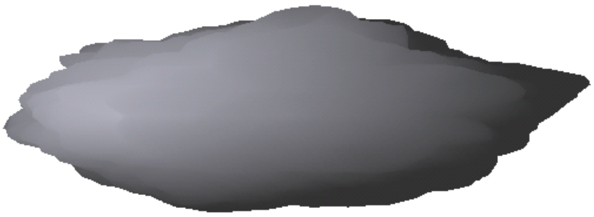
Second midpoint-displacement step



Second subdivision damping step



re-scaled model



re-scaled with diffuse light

textured with diffuse light

bump-mapped and textured

References

[1] [L. C. Carpenter, *Computer Rendering of Fractal Curves and Surfaces*, SIGGRAPH 1980.

[2] H.-O. Peitgen and D. Saupe, *The Science of Fractal Images*, pages 243 - 260, Springer-Verlag, New York, 1988.

[3] P. Prusinkiewicz, M. Hammel, *A Fractal Model of Mountains with Rivers*, Proceeding Graphics Interface 1993.

[4] F. K. Musgrave, *Methods for Realistic Landscape Imaging*, Dissertation Yale University 1993.

[5] A. R. Smith, *Plants, Fractals and Formal Languages*, Proceeding SIGGRAPH 1984.

[6] E. Catmull, J. Clark, *Recursively Generated (B)-Spline Surfaces on Arbitrary Topological Meshes*, 1978, Computer-Aided Design, Volume 10, pages 350 - 355.

[7] B. B. Mandelbrot, *The Fractal Geometry of Nature*, 1977, W. H. Freeman and Company.

[8] H.-O. Peitgen, J. Hartmut, D. Saupe, *Chaos and Fractals: New Frontiers of Science*, 1992, Springer-Verlag.

[9] T. Möller, E. Haines, *Real-Time Rendering*, 1999, 1st Edition, A K Peters, Ltd.

[10] R. L. Cook, *Shade Trees*, 1984 Computer Graphics SIGGRAPH ’84 Proceedings Vol. 18, pages 223 – 231.

[11] J. Kautz, H.-P. Seidel, *Hardware Accelerated Displacement Mapping for Image Based Rendering*, Graphics Interface ’01.

[12] D. R. Peachey, *Solid Texturing of Complex Surfaces*, SIGGRAPH ’85 Proceedings, pages 279 – 286.

[13] N. A. Carr, J. C. Hart, *Real-Time Procedural Solid Texturing*, 2001, SIGGRAPH ’01, chapter 13, and SIGGRAPH ’02, chapter 6.