Point Evaluation of Multi-Variable Random Fractals

Dietmar Saupe
Institut für Dynamische Systeme
Fachbereich Mathematik
Universität Bremen
2800 Bremen-33

Abstract

We present a method to compute a random fractal function in 1,2,3 or more variables. It blends techniques from the Mandelbrot-Weierstrass function and Perlin's turbulence function (used for texturing). With this approach we may prescribe fractal dimension and lacunarity as parameters, which are even allowed to vary as in a continuous function. Further the method provides pointwise evaluation of the functions as opposed to displacement techniques which require global computation, and in many cases it is superior to other methods in speed. As an illustrative example a fractal planet with continents of differing fractal dimensions is given.

Eine Methode zur Berechnung stochastischer fraktaler Funktionen in 1,2,3 oder mehr Variablen wird vorgestellt. Dazu werden bekannte Techniken der Mandelbrot-Weierstrass-Funktion und der Turbulenz-Funktion von Perlin (die für Texturen entwickelt wurde) kombiniert. Durch diesen Ansatz gelingt es, fraktale Dimension und Lakunarität als Parameter der Methode einzuführen. Diese können sogar wie stetige Funktionen variieren. Der Algorithmus kann die generierte Funktion punktweise auswerten, ohne daß das Fraktal schon global berechnet ist, wie es bei den Displacement-Methoden notwendig ist. In Anwendungen ist die vorgestellte Methode häufig effizienter als andere Verfahren. Als illustratives Beispiel für die Methode wird ein fraktaler Planet mit Kontinenten unterschiedlicher fraktaler Dimension gezeigt.

1 Introduction

Random fractals as introduced by B. Mandelbrot in *The Fractal Geometry of Nature* [4] have enjoyed a wide popularity in the computer graphics community. A number of algorithms for their generation have been proposed, see e. g. [1,2,5,9,10]. Let us first recall some definitions. A random fractal is a multi-variable random function $V(x)$, $x = (x_1, ..., x_n)$ such that

- The increments $V(y) - V(x)$ are Gaussian with mean zero.
- The variance of the increments $V(y) - V(x)$ depends only on the Euclidean distance $||y - x||$. For a number $0 < H < 1$ we have

$$E(|V(y) - V(x)|^2) \propto ||y - x||^{2H}$$

where E denotes expectation.

The function V has stationary increments and is isotropic, i. e. all points and all directions are statistically equivalent. In the frequency domain we have that the spectral density falls off as $1/f^{2H+n}$. The fractal (box counting) dimension of the graph of V is $D = n + 1 - H$.

Random fractals can be regarded as models for a large number of natural phenomena, the most prominent ones being landscapes. Here $V = V(x, y)$ is a function of two variables and interpreted as height at position (x, y). A function $V = V(x, y, z)$ of three variables may specify e. g. water vapor density in a volume giving models of clouds.

A technique very much related to random fractals has been introduced by Perlin in 1985 [8]. In his framework he considers random functions of three variables as models of *solid texture*. The functions are generated procedurally and the approach has been termed *functional-based modelling*.

In this note we suggest to blend Perlin's techniques with some stemming from random fractals, in particular the Mandelbrot-Weierstrass function. As a result we obtain a simple and fast method which has some striking advantages over previously known algorithms for the generation of random fractals:

- The fractal dimension may be specified. We can even let the dimension D be a function $D = D(x_1, ..., x_n)$ varying in space.
- Lacunarity may also be specified globally or as a function.
- Although a random fractal function at a given point generally depends on its values at all other points the calculation is a point evaluation and no explicit references to other points are necessary. Therefore the order of the computation does not matter which is important for implementations on parallel processors.
- The method is relatively simple to implement and still offers a significant increase in speed for many cases of interest.

This paper is organized as follows. In the second section we describe the approach of Perlin for random fractal functions. In the next section we introduce two parameters to the

method, which govern the fractal dimension and the lacunarity. This is motivated by the Mandelbrot-Weierstrass function. In sections 4 and 5 we make some implementation notes and compare methods with respect to computational cost. As an illustrative example we derive a fractal planet with continents of differing fractal dimensions.

2 A first multidimensional noise function

The goal of the construction is a random function $V_3(x, y, z)$ of three variables, which takes real values. All points (x, y, z) and all directions should be statistically equivalent. Moreover, in the frequency domain the spectral density function is of the form $1/f^\beta$, where f denotes the combined spatial frequencies in the x-, y- and z-direction, i. e. $f = \sqrt{f_x^2 + f_y^2 + f_z^2}$.

Let us begin with a one-dimensional model $V_1(x)$. First an auxiliary function $S_1(x)$ is defined at integer points $k \in \mathbf{Z}$: The value $S_1(k)$ is taken as a sample of a random variable with mean zero (e. g. a Gaussian random variable). Then a differentiable, interpolating function $S_1(x)$ is constructed from the data. Here one can choose piecewise cubic Hermite interpolation with the additional constraints $S_1'(k) = 0$ for integers k (see [6] and below). Other choices are possible, of course. in [8] the auxiliary function S_1 is called "noise" . But S_1 contains primarily low frequencies, and therefore should not conceived as $1/f$-noise. However, as we sum up scaled-down copies of S_1 we obtain a $1/f$-noise. More precisely, the one-dimensional noise function is defined as

$$V_1(x) = \sum_{k=0}^{\infty} \frac{1}{2^k} S_1(2^k x). \tag{1}$$

This is very similar to the original *turbulence* function used in [8] i. e. $\sum_{k=0}^{\infty} 2^{-k} |S_1(2^k x)|$. As k grows, the contributions to the sum rapidly get small, thus, in practice the summation is carried out only over a few terms. Due to the construction of $S_1(x)$ as interpolated random data at integer sites we expect $S_1(x)$ to exhibit a power spectrum which has only little contributions at frequencies above 1. The power spectrum of $V_1(x)$ can be derived from the spectrum of $S_1(x)$ and we expect a $1/f^\beta$ power law. In fact, the exponent β turns out to be equal to 3 (see figure 2 and next section). Thus, $V_1(x)$ is not a model for Brownian motion as claimed in [8] since Brownian motion obeys a $1/f^2$ power law [10] (the squared amplitudes fall off as $1/f^2$).

The generalization to two, three or more dimensions is straightforward. E. g. in the 2-dimensional case we define an auxiliary function $S_2(x, y)$ first at integer lattice points (k, l) again by sampling a random variable. Secondly, the data is interpolated by a smooth function. The noise function becomes

$$V_2(x, y) = \sum_{k=0}^{\infty} \frac{1}{2^k} S_2(2^k x, 2^k y). \tag{2}$$

Clearly, $V_2(x, y)$ is a continuous function of x and y. However, we point out that not all directions in the xy-plane are equivalent. On the x- and y-axis one random data enters per unit interval, whereas e. g. on the diagonal, the distance between random data points is $\sqrt{2}$.

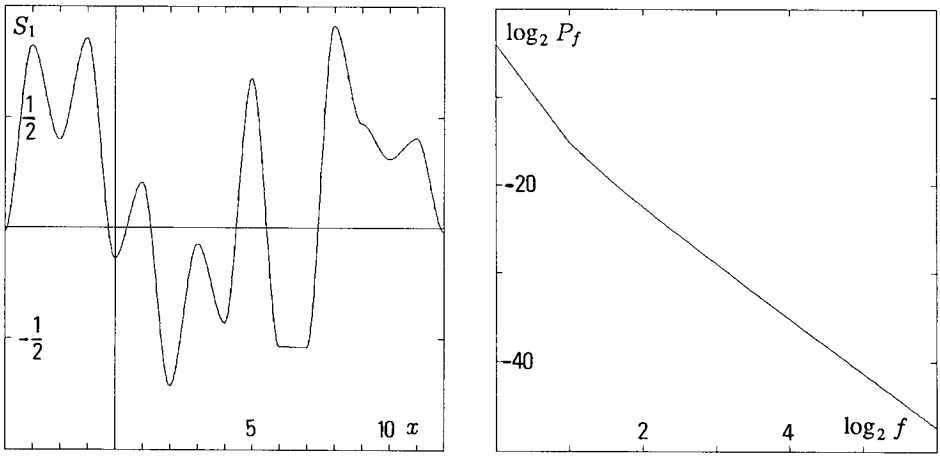

Figure 1. A sample of an auxiliary function $S_1(x)$ (left) and its spectral density (right) on a doubly logarithmic (base 2) scale. The spectral density plot is obtained from averaging squared amplitudes of several periodograms of samples of $S_1(x)$. Since $S_1(x)$ is two times continuously differentiable with a piecewise continuous third derivative, we expect that the amplitudes fall off as $1/f^3$. The spectral density function should obey a $1/f^6$ power law. In fact, the slope of the graph is about -6.

This necessarily shrinks the width of the power spectral density of $S_2(x, y)$: The power spectrum of $V_2(x, y)$ decays by a factor of about $\sqrt{2}$ faster along the diagonal. In all of our experiments this lack of purity has not been visually noticeable.

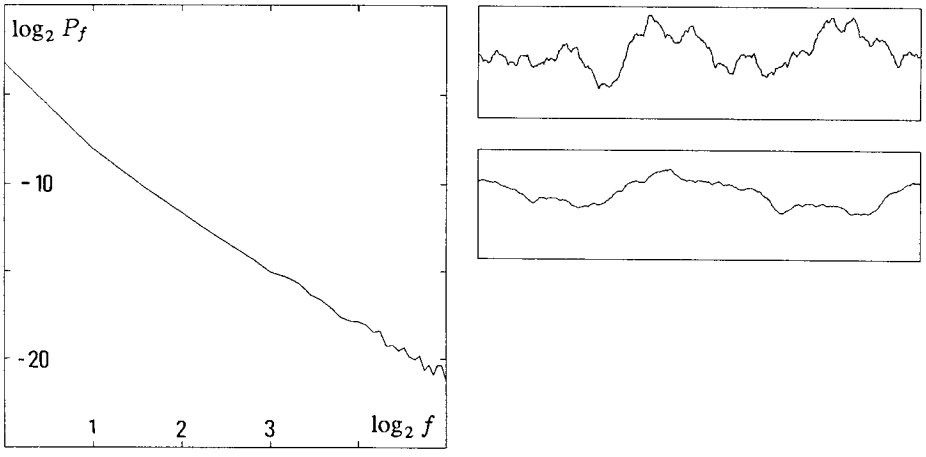

Figure 2. Power spectrum of V_1 on a base 2 logarithmic scale (left) and a sample of $V_1(x)$ (top right) where x varies over 10 units. The bottom graph shows a detail (the first tenth) of the above graph. Note that the slope of the power spectrum is about -3 corresponding to a $1/f^3$ power law.

In the 3-dimensional case, which is the most interesting case in applications we thus define the noise function $V_3(x, y, z)$

$$V_3(x, y, z) = \sum_{k=0}^{\infty} \frac{1}{2^k} S_3(2^k x, 2^k y, 2^k z). \tag{3}$$

The interpolation suggested in [6,8] is as follows : Consider a point (x, y, z) in 3-space. Let us write

$$x = i_x + d_x, \quad y = i_y + d_y, \quad z = i_z + d_z$$

where i_x, i_y, i_z are integers and d_x, d_y, d_z are nonnegative fractions less than 1. Then set

$$s_x = d_x^2(3 - 2d_x), \quad s_y = d_y^2(3 - 2d_y), \quad s_z = d_z^2(3 - 2d_z)$$

and define

$$\begin{aligned} S_3(x,y,z) \;=\; & s_x s_y s_z \, S_3(i_x+1, i_y+1, i_z+1) \\ & + (1-s_x) s_y s_z \, S_3(i_x, i_y+1, i_z+1) \\ & + s_x(1-s_y) s_z \, S_3(i_x+1, i_y, i_z+1) \\ & + (1-s_x)(1-s_y) s_z \, S_3(i_x, i_y, i_z+1) \\ & + s_x s_y(1-s_z) \, S_3(i_x+1, i_y+1, i_z) \\ & + (1-s_x) s_y(1-s_z) \, S_3(i_x, i_y+1, i_z) \\ & + s_x(1-s_y)(1-s_z) \, S_3(i_x+1, i_y, i_z) \\ & + (1-s_x)(1-s_y)(1-s_z) \, S_3(i_x, i_y, i_z) \end{aligned} \tag{4}$$

This piecewise cubic interpolation yields continuously differentiable real function in $\mathbf{R}^3$.

3 Fractal dimension and lacunarity

The *turbulence* function in [8] and also the noise function of the preceding section does not have any parameters. It is desirable, however, to control the power spectrum of the noise in order to be able to choose the fractal dimension and the lacunarity of the generated noise. These two parameters can best be described in the spectral domain as in figure 3. Let us consider the one-dimensional case first. If the spectral density falls off as $1/f^{2H+1}$ with $0 \leq H \leq 1$, then the fractal dimension of the graph of the random fractal is $D = 2 - H$ (see [10]). Lacunarity adds a certain texture to the noise function without changing the fractal dimension, see [10].

One method to create a random function with given fractal dimension and lacunarity is by the Mandelbrot-Weierstrass function ([10])

$$V_{MV}(x) = \sum_{k=-\infty}^{\infty} \frac{A_k}{r^{kH}} \sin(2\pi r^k x + \phi_k) \tag{5}$$

where A_k is a Gaussian random variable with the same expectation and variance for all k. ϕ_k is a uniformly distributed random variable with values in the interval $[0, 2\pi]$ and r is a scaling factor larger than 1. As k grows, terms of higher frequencies but lower amplitudes are added to the sum. Following the discussion in [10] we have contributions of discrete frequencies $f_k = r^k$ and of mean square amplitude proportional to $1/r^{2kH}$. Thus, the power spectral density, computed as the mean square amplitudes divided by the bandwidth, is proportional to $1/f_k r^{2kH}$ which is equal to $1/f_k^{2H+1}$. Thus, the fractal dimension of the graph of $V_{MV}(x)$ is $2 - H$.

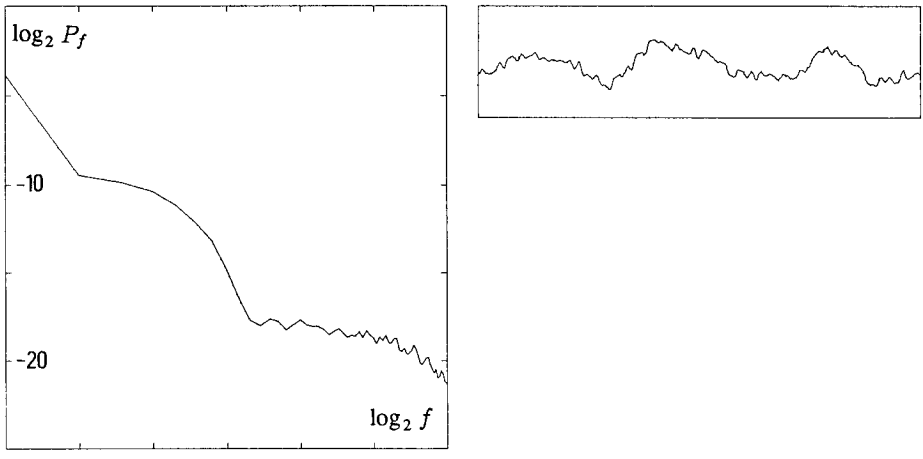

Figure 3. Power spectrum (left) for $1/f$-noise with lacunarity (right). The absolute value of the overall slope is equal to the exponent in the $1/f^{\beta}$ power law of the generated noise. The lacunarity is indicated as oscillating amplitude of the power spectrum. This plot has been obtained for $V_1(x)$ of equation (6) with $r = 10$ and $H = 0.5$.

Comparing the Mandelbrot-Weierstrass function with the sum $\sum_{k=0}^{\infty} 2^{-k} S_1(2^k x)$ we see similarities and differences. In both approaches, auxiliary functions with increasing frequencies and decreasing amplitude are summed up. The differences are as follows :

- The sine function used in $V_{MV}(x)$ has a point spectrum (all power at frequency 1), whereas the other function $S_1(x)$ has its power distributed over a finite bandwidth.
- The randomization in $V_{MV}(x)$ is by means of the random amplitude and phase. On the other hand $S_1(x)$ already contains randomness by construction.
- In $V_{MV}(x)$ the rate at which the contributing frequencies fall off is controlled by $r > 1$. The larger the parameter r, the greater is the resulting lacunarity of the fractal. The corresponding factor in Perlin's formula is fixed at $r = 2$.
- Also the decay rate of the amplitudes of the terms in the summation are different. In $V_{MV}(x)$ it is $1/r^{kH}$, $0 \leq H \leq 1$ versus $1/r^k$ (with $r = 2$) in $V_1(x)$. The parameter H controls the fractal dimension of the graph of $V_{MV}(x)$.
- At last we note that the summation in $V_{MV}(x)$ is carried out from $-\infty$ to ∞. The extra small frequencies are added to ensure statistical selfsimilarity not only at small scales but also at all arbitrarily large scales. Thus, when we sum only from 0 to ∞ we must be careful in that we only apply the model to objects which are of size of order 1 or smaller.

Motivated by $V_{MV}(x)$ we introduce corresponding parameters governing fractal dimension and lacunarity of $V_1(x)$. We redefine

$$V_1(x) = \sum_{k=-\infty}^{\infty} \frac{1}{r^{kH}} S_1(r^k x) \tag{6}$$

where $r > 1$ determines lacunarity and $H > 0$ controls the fractal dimension D of the graph of $V_1(x)$. Although the situation is somewhat different from the Mandelbrot-Weierstrass function — the power spectrum of $S_1(x)$ is not a point spectrum — a theoretical result verifies $D = 2 - H$ again. This has recently been shown by Kaplan, Mallet-Paret and Yorke [3]. In fact, for a large class of functions $S_1(x)$ the fractal dimension of the graph of $V_1(x)$ is equal to $2 - H$ independently of $r > 1$. E. g. all finite trigonometric sums

$$\frac{a_0}{2} + \sum_{k=1}^{m} a_k \cos kt + b_k \sin kt$$

are included. In actual applications as shown in section 4 one works with a periodic function $S_1(x)$ with a large period, say 50 or 100, which can be closely approximated by a Fourier sum of the above form. Therefore it is safe to assume, that here we also have

$$D = 2 - H$$

as the fractal dimension of the graph of $V_1(x)$, independently of $r > 1$. In the one-dimensional example of last section we have the special case of (6) with $r = 2$ and $H = 1$. Therefore a $1/f^3$ power law holds and the graph of $V_1(x)$ has a dimension $D = 1$, i. e. it is a borderline fractal.

Let us employ numerical techniques to check the relationship between H and D as follows. We can compute the fractal dimension of the graph of a sample of $V_1(x)$ from the length of the graph measured on different scales. More precisely, let the length of the graph of $V_1(x)$, $0 < x < 1$ measured on the scale $\Delta x > 0$ be given as

$$L(\Delta x) = \sum_{k=0}^{1/\Delta x - 1} |V_1((k+1)\Delta x) - V_1(k\Delta x)|.$$

Then (see [10])

$$L(\Delta x) \propto \frac{1}{\Delta x^{D-1}}$$

and with $L_k = L(2^{-k})$ we get

$$D = \lim_{k \to \infty} \left(1 + \frac{\log L_k / L_{k-1}}{\log 2}\right).$$

With this approach we empirically determine table 1. As expected, the result confirms the formula $D = 2 - H$.

Let us summarize the result for random fractal evaluation in multiple dimensions as follows :

Rescale-and-add method : *Let $S_n : \mathbf{R}^n \to \mathbf{R}$ be a real function that has values at integer lattice points of $\mathbf{R}^n$ defined by Gaussian random variables of zero mean and the same variance at all points. Further let $S_n(x_1, \ldots, x_n)$ be a smooth interpolation from the data at the integer lattice points. Then the function $V_n : \mathbf{R}^n \to \mathbf{R}$ defined by*

$$V_n(x) = \sum_{k=k_0}^{\infty} \frac{1}{r^{kH}} S_n(r^k x) \tag{7}$$

	$H = 0.2$					
	$r = \sqrt{2}$		$r = 2$		$r = 4$	
k	L_k	D_k	L_k	D_k	L_k	D_k
8	141.9	1.68	112.2	1.79	90.6	1.92
9	237.4	1.74	200.3	1.84	150.3	1.73
10	417.2	1.81	350.0	1.80	284.7	1.92
11	748.7	1.84	613.0	1.81	460.3	1.69
12	1309.3	1.81	1069.9	1.80	862.5	1.91
13	2275.9	1.80	1865.8	1.80	1390.8	1.69
	$H = 0.5$					
	$r = \sqrt{2}$		$r = 2$		$r = 4$	
k	L_k	D_k	L_k	D_k	L_k	D_k
8	23.6	1.40	19.5	1.45	14.8	1.76
9	33.1	1.49	27.7	1.51	17.8	1.26
10	46.6	1.49	39.6	1.52	30.5	1.78
11	66.6	1.51	56.1	1.50	36.5	1.26
12	93.7	1.49	79.6	1.50	61.2	1.74
13	133.5	1.51	113.4	1.51	72.7	1.25
	$H = 0.8$					
	$r = \sqrt{2}$		$r = 2$		$r = 4$	
k	L_k	D_k	L_k	D_k	L_k	D_k
8	6.2	1.17	4.7	1.21	3.3	1.44
9	7.2	1.20	5.3	1.18	3.5	1.07
10	8.2	1.20	6.3	1.25	4.4	1.35
11	9.4	1.20	7.2	1.20	4.7	1.08
12	10.8	1.20	8.3	1.21	6.0	1.34
13	12.5	1.20	9.6	1.21	6.3	1.08

Table 1 . Computation of fractal dimensions of graphs of $V_1(x)$ with varying parameters H and r. The columns D_k list the quantities $1 + \frac{\log L_k/L_{k-1}}{\log 2}$. The expected dimensions are 1.8, 1.5, 1.2 for $H = 0.2, 0.5, 0.8$ respectively. The values of the table meet these expectations except maybe for the case $r = 4$, where we obtain the right dimensions when we average two consecutive numbers D_k. Graphs of corresponding random fractal functions and power spectra for $r = \sqrt{2}, 4$ are shown in figure 4.

with $r > 1, 0 \leq H \leq 1$ *and* $k_0 \leq 0$ *is a random function whose graph has a fractal dimension*

$$D = n + 1 - H \tag{8}$$

and $r > 1$ *determines lacunarity.*

4 Implementation and comparison with other methods

The implementation of formula (7) is not hard. One of the two issues that must be considered is the range of the summation. The lower index k_0 should obviously be chosen small enough so that the largest scale L of the objects is not greater that r^{-k_0}, or, i. e.

$$r^{k_0} L \leq 1. \tag{9}$$

If k_0 is too big one sees the effect of the dominant frequencies (≈ 1) of the auxiliary function S_n. The cutoff at the other end of the summation is harder to define. If the random function

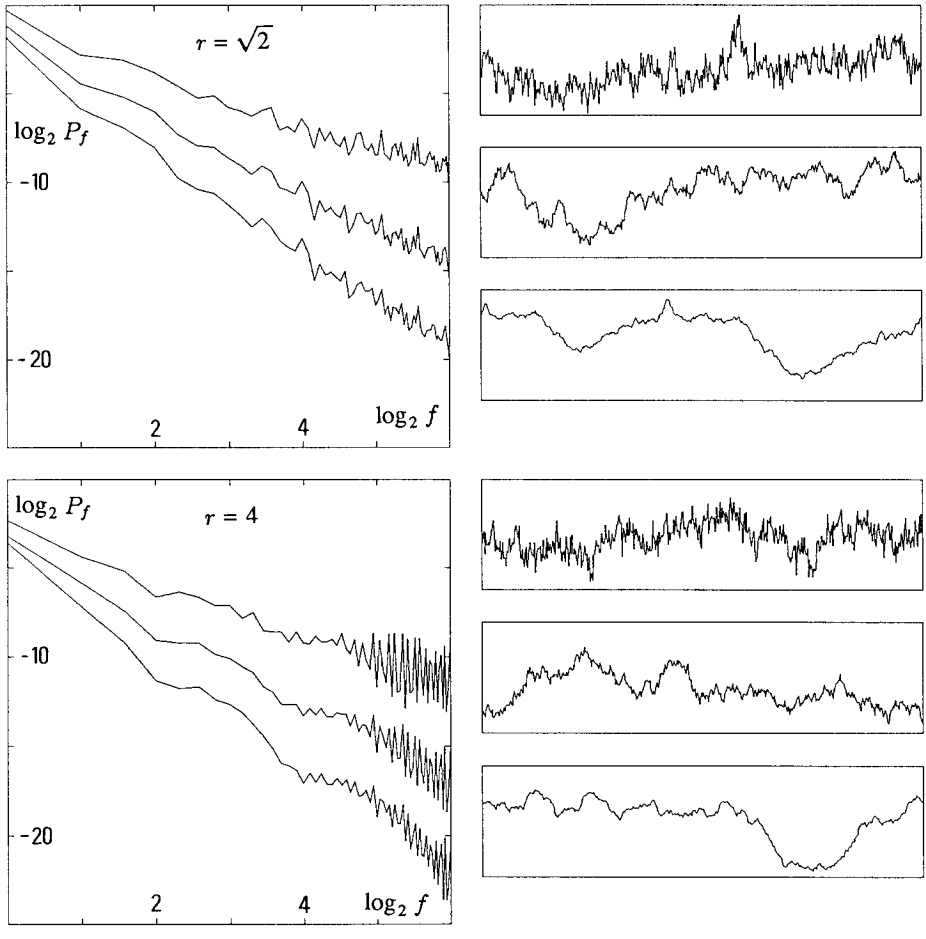

Figure 4. Power spectra and noise samples for different parameters of the rescale-and-add method. The lacunarity parameter r in the top graphs is $r = \sqrt{2}$, in the bottom graphs $r = 4$. Curves for three values of H are shown: $H = 0.2$ (top), $H = 0.5$ (center) and $H = 0.8$ (bottom). The corresponding exponent β of the $1/f^{\beta}$ power law are $\beta = 1 + 2H$, i. e. $\beta = 1.4$ (top), $\beta = 2.0$ (center) and $\beta = 2.6$ (bottom). These are actually the approximate negative slopes of the empirically computed power spectra as shown. On the right corresponding random fractal functions (samples of $V_1(x)$) are given. The fractal dimensions are checked numerically in Table 1.

will be sampled at points which have a distance Δ from each other, then we must be aware of aliasing effects if the summation is carried out with terms exceeding the Nyquist limit, i. e. terms with $r^k > 2/\Delta$. This motivates a cutoff at

$$k \approx \frac{\log 2/\Delta}{\log r}. \tag{10}$$

The corresponding discussion of Peachey in [6] also applies here. However, if H is small, then the amplitudes of the clamped terms may still be very large and an aesthetically more pleasing picture may be obtained if the summation in (7) is continued for a few more terms than suggested by (10).

The other remaining problem is the production of the random numbers attached to the integer lattice points. The approach by Peachey [6] suggests to work with a table $T[k]$, $k = 1, ..., N-1$ of a few (say 100) random numbers. A lattice point with integer coordinates (x, y, z) is assigned a value V of the table via

$$V(x, y, z) = T[|x + 3y + 5z| \bmod N].$$

This method is not satisfactory because the results in planes $x + 3y + 5z = \text{const}$ will not be independent. In fact, the repetitive patterns are easily detected by the eye.

We suggest to invest in terms of storage space by letting T be a three-dimensional array

$$T[i, j, k],\ i, j, k = 0, ..., N-1.$$

For given integers x, y, z we define integers i_x, i_y, i_z as in

$$i_x = \begin{cases} x \bmod N & \text{if } x \geq 0 \\ x \bmod N + N & \text{if } x < 0 \end{cases}$$

Then we let $V(x, y, z) = T[i_x, i_y, i_k]$.

An alternative method would be to procedurally define $V(x, y, z)$ for integers x, y, z. One may use the combined bits of x, y, z as a state or seed for one of the usual random number generators. $V(x, y, z)$ could then be taken as the result of several consecutive calls to the random number generator. One call is certainly not enough, since neighboring lattice points often differ only by a single bit and this difference is not sufficiently enlarged by a single call. The cost of this procedure is rather high as compared to the table lookup approach from above which obviously yields acceptable results.

5 Comparison with other methods

There are at least two other methods which allow control of dimension and lacunarity. The first one is related to the well known midpoint displacement methods, it has been termed *random successive additions*, see [9,10]. The other method uses the Mandelbrot-Weierstrass function in several variables. In the following we briefly describe both methods. Then we continue to point out the differences between these methods and the rescale-and-add method. Finally we estimate and compare the costs of the algorithms.

5.1 Random additions : displacing interpolated points

Let us assume, that a random function $V(x, y, z)$ with prescribed fractal dimension D and lacunarity r for a final resolution of N^3 points has to be generated. If already M^3, $M < N$ points are calculated, then in the next stage a total of $(rM)^3$ points are determined in two steps ($r > 1$).

1. The old points are interpolated (e. g. in a multilinear fashion) at $(rM)^3$ points.

2. A displacement is added to all points. At each point the displacement is determined by sampling a Gaussian random variable with mean zero and the same variance for all points. Let us remark that the method can be adapted to handle locally changing fractal dimensions by letting the variance of the displacements depend on the local dimension.

These steps are repeated until a resolution of N^3 points is achieved. For details and pseudo-code see [9].

5.2 Mandelbrot-Weierstrass method

To generalize equation (5) to three dimensions we set

$$V_{MV}(x,y,z) = \sum_{i,j,k=-\infty}^{\infty} \frac{A_{i,j,k}}{r^{(i+j+k)H}} \cdot \sin(2\pi r^i x + \phi_{1,i}) \cdot \sin(2\pi r^j y + \phi_{2,j}) \cdot \sin(2\pi r^k z + \phi_{3,k}) \tag{11}$$

where $A_{i,j,k}$ are Gaussian random variables with the same variance and $\phi_{1,i}, \phi_{2,j}, \phi_{3,k}$ are random phases in $[0, 2\pi]$. In practice, the summation is carried out over finitely many terms determined by the largest and smallest scale of the underlying object. E.g, , if (x,y,z) is restricted to the unit cube, then it is sufficient to start the summation with $i,j,k = 0$.

5.3 Comparison

The major difference between the successive addition and the other methods lies in the fact, that in the first method the complete fractal must be computed before any value of $V(x,y,z)$ can be given. This implies rather stringent requirements regarding storage capacities. A sample of $V_3(x,y,z)$ covering the unit cube at a resolution of 1000 points per side will devour space for 10^9 floating point numbers. In summary we have so far that this method is impractical when applied to volumes as opposed to two-dimensional domains.

In order to estimate the cost for the evaluation of $V_3(x,y,z)$ for all methods, let us assume as above that we compute the fractal function in the unit cube up to a resolution of $1/N$. For the interpolation method with random successive additions this implies that we have to go through $m = \log N / \log r$ stages to achieve the final resolution. For the other methods we sum the auxiliary functions up to frequencies $N/2$. Thus, here the involved sums consist of $m = \log N / \log r$ terms. In the interpolation method a total of about

$$N^3 + \left(\frac{N}{r}\right)^3 + \left(\frac{N}{r^2}\right)^3 + \ldots + \left(\frac{N}{r^{n-1}}\right)^3 = N^3 \cdot \frac{1 - r^{-3m}}{1 - r^{-3}}$$

interpolations and random offsets must be performed. The interpolations cost about 27 floating point operations (flops) each, and let us assume that one random addition can be done in 15 flops (includes a few calls to a random number generator with averaging to obtain a Gaussian random number with correct variance). Thus, the total cost averaged per point is about

$$42 \frac{r^3}{r^3 - 1} \text{ flops.}$$

In the three-dimensional Mandelbrot-Weierstrass method we obtain one evaluation of the random fractal $V_3(x,y,z)$ in

$$3\left(\frac{\log N}{\log r}\right)^3 \text{ flops.}$$

Here the cost for the generation of the random coefficients and phases as well as for the table lookups of sine values is neglectable (at least if $V_3(x,y,z)$ has to be evaluated at many points from a grid).

In our rescale-and-add method, formula (7) with $n = 3$, we arrive at $5m$ flops plus m evaluations of S_3 each one costing roughly 60 flops. The result is a total of

$$65\frac{\log N}{\log r} \text{ flops.}$$

Let us illustrate these numbers by choosing $N = 2^{10} = 1024$ and $r = 2$. Then we have for the approximate costs in the three-dimensional case

method (3D)	cost/point	$r = 2,\ N = 1024$
method of successive additions	$42r^3/(r^3-1)$	48 flops / point
Mandelbrot-Weierstrass function	$3(\log N/\log r)^3$	3000 flops / point
rescale-and-add method	$65 \log N/\log r$	650 flops / point

We see that the ability of the lower two methods to compute values of $V_3(x,y,z)$ at individual points has its price. However, these two methods will still be more economical when only surfaces in 3-space are of interest, which seems to be the case most interesting in applications. In this sense we have that the rescale-and-add method is the most economical method in 3-space. In flat two-dimensional space we obtain the following table of costs :

method (2D)	cost/point	$r = 2,\ N = 1024$
method of successive additions	$26r^2/(r^2-1)$	35 flops / point
Mandelbrot-Weierstrass function	$3(\log N/\log r)^2$	300 flops / point
rescale-and-add method	$31 \log N/\log r$	310 flops / point

Here the difference between the Mandelbrot-Weierstrass and the rescale-and-add method is insignificant. If $V_2(x,y)$ has to be computed on a grid of N^2 points, then the method of successive random additions is the method of choice.

6 Fractal planet application

Perlin already has demonstrated the many uses of the auxiliary functions S_k and of his derived *turbulence* procedure which is almost the same as $V_3(x,y,z)$ with $r = 2$ and $H = 1$ fixed. Here we give only one more example. Imagine the evaluation of an instance of $V_3(x,y,z)$ for points (x,y,z) on the unit sphere. Regard these values as height above or below sealevel,

and already we have a model for a fractal planet. The remaining – still difficult – part is the rendering with the proper choice of colors.

The difference between the well known fractal planet of Voss (see [4] and [10]) and this one is that his coastlines necessarily have fractal dimension 1.5 by construction (using the method of random cuts) whereas here we have both, the fractal dimension and the lacunarity, under control. In our color figure the fractal dimension of the coastlines ranges from 1.2 near the equator to 1.9 at the poles, which are only partly visible since they are covered by "ice". Of course, a similar fractal planet with parameters can be derived by the Weierstrass-Mandelbrot function as explained in section 4.

References

[1] Fournier, A. , Fussell, D. and Carpenter, L., *Computer rendering of stochastic models*, Comm. of the ACM 25 (1982) 371–384

[2] Lewis, J.P., *Methods for stochastic spectral synthesis*, ACM Transactions on Graphics (1987)

[3] J.L. Kaplan, J. Mallet-Paret, J.A. Yorke, *The Lyapunov dimension of a nowhere differentiable attracting torus*, Ergod. Th. and Dynam. Sys. 4 (1984) 261–281

[4] Mandelbrot, B.B., *The Fractal Geometry of Nature*, W.H.Freeman and Co., New York, 1982

[5] Miller, G.S.P., *The definition and rendering of terrain maps*, Computer Graphics 20,4 (1986) 39–48

[6] Peachey, D.R., *Solid textures and anti-aliasing issues*, in : Functional-based modelling, SIGGRAPH'88 Course Notes 28, Atlanta 1988

[7] Peitgen, H.-O. and Saupe, D. (eds), *The Science of Fractal Images*, Springer-Verlag, New York, 1988

[8] Perlin, K., *An image synthesizer*, Computer Graphics 19,3 (1985) 287–296

[9] Saupe, D., *Algorithms for random fractals*, in : The Science of Fractal Images, Peitgen, H.-O. and Saupe, D. (eds), Springer-Verlag, New York, 1988

[10] Voss, R.F., *Fractals in nature : From characterization to simulation*, in : The Science of Fractal Images, Peitgen, H.-O. and Saupe, D. (eds), Springer-Verlag, New York, 1988