A Multiple-Scale Stochastic Modelling Primitive

Jos Stam
Eugene Fiume

Department of Computer Science
University of Toronto
10 King's College Road
Toronto, Canada, M5S 1A4

Abstract

Stochastic modelling has been successfully used in computer graphics to model a wide array of natural phenomena. In modelling three-dimensional fuzzy or partially translucent phenomena, however, many approaches are hampered by high memory and computation requirements, and by a general lack of user control. We will present a general stochastic modelling primitive that operates on two or more scales of visual detail, and which offers considerable flexibility and control of the model. At the macroscopic level, the general shape of the model is constrained by an ellipsoidal correlation function that controls the interpolation of user-supplied data values. We use a technique called *Kriging* to perform this interpolation. The microscopic level permits the addition of noise, which allows a user to add interesting visual textural detail and translucency. A wide variety of noise-synthesis techniques can be employed in our model. We shall describe the mathematical structure of our model, and give an attractive rendering implementation that can be embedded in a traditional ray tracer rather than requiring a volume renderer. As an example, we shall apply our approach to the modelling of clouds.

Résumé

En infographie, nombreux phénomènes naturels ont été simulés de manière convainquante par des modèles stochastiques. Néanmoins, dans la cas de phénomènes tri-dimensionels partiellement translucides ou flous, la plupart de ces modèles sont très friands en mémoire et en temps machine, et n'offrent qu'un contrôle limité du modèle à l'utilisateur. Dans cet article nous présenterons un modèle stochastique général opérant sur deux ou plusieurs niveaux de détail visuel, qui est facilement controlable par l'utilisateur. Au niveau macroscopique, la forme générale du modèle est une interpolation de données, spécifiées par l'utilisateur, soumise à une fonction de corrélation ellipsoidale. Nous utilisons une technique appelée *Krigage* pour l'interpolation. Le niveau microscopique permet l'addition de bruit, permettant à l'utilisateur d'ajouter une texture visuelle intéressante et une transparence au modèle. A ce niveau, un grand nombre de techniques de synthèse de bruit peuvent être utilisées. Nous décriverons la structure mathématique de notre modèle et présenterons une mise en oeuvre d'un algorithme de synthèse d'images de notre modèle, qui peut être facilement incorporée dans un logiciel standard: le lancé de rayons. Comme exemple d'utilisation, nous appliquerons notre modèle à la simulation de nuages.

Keywords: stochastic modelling, simulation of clouds, scattered data interpolation, solid textures, fractals, ray tracing.

1 Introduction

Many kinds of natural phenomena are resistant to direct deterministic physical or geometric modelling. A physical model, assuming one exists, can be too costly to compute, while a geometric model can be too large to manipulate efficiently. Hence it is appropriate to search for *visual models* instead. This means a model that simulates the perceived behaviour of the phenomenon.

Our concern in this paper is the modelling of objects that have a discernible shape and have nonuniform density or opacity. Among others, clouds, fire, and various classes of texture fall into this category. The model presented in this paper is analytical, it has the advantages of having low storage requirements, and it is easily incorporated into standard rendering software (such as a ray tracer). A user controls both the global shape and the small scale detail of the phenomenon by specifying a correlation structure. Interestingly, the model turns out to be a generalization of Blinn's "blobbies" [4] and Gardner's textured ellipsoids [7]. As a case study, we shall apply our model to simulate clouds. Clouds are interesting because of the wide variety of shapes and visual effects they exhibit. The next section reviews the basic notions and notations of stochastic modelling, and Section 3 reviews related work in computer graphics. Section 4 informally presents our model, the mathematics of which is discussed in Sections 5 and 6. Rendering issues are discussed in Section 7, followed by some basic modelling results in Section 8.

2 Stochastic Modelling

One conceptually simple approach to modelling a natural phenomenon is to specify it completely by a large set of primitives such as polygons or particles. This set can be generated either by the user, which is somewhat impractical, or by an algorithm. Reeves has successfully used particles to model fire and grass [18]. Rendering cost makes it difficult to use particles for other phenomena. See, for example, the interesting but computationally-prohibitive approach taken by Kajiya and Von Herzen to model and render clouds [9].

Another approach is to model a natural phenomenon as a function $R(x, y, z)$. For example, in the case of clouds the function could be the density at a given location of space, with a range of values between one, denoting total opacity, and zero, denoting total translucence. Many choices for R are possible, the most common being smooth surfaces such as splines or blobbies [4]. More generally R could be any mathematical function, but to capture the irregularity of many natural phenomena, a stochastic function is often employed [6].

2.1 Random Fields

A multi-dimensional phenomenon can be modelled as a *random field* R. At each location $\mathbf{t} \in \mathbb{R}^d$, a *random variable* $R(\mathbf{t})$ is characterized by a probability distribution:

$$F_{\mathbf{t}}(r) = Prob(R(\mathbf{t}) \leq r). \tag{1}$$

In practice, this distribution is unknown or nonexistent. Another way to characterize a random process is to specify its statistics. The two best-known statistics are the *mean* $\mu(\mathbf{t}) = E[R(\mathbf{t})]$, and the *variance* $\sigma^2(\mathbf{t}) = E[R(\mathbf{t})^2] - \mu(\mathbf{t})^2$, where

$$E[g(R(\mathbf{t}))] = \int_{\mathbb{R}} g(r)\, dF_{\mathbf{t}}(r). \tag{2}$$

In the simplest case, these two values are independent of $\mathbf{t}$, giving rise to a *homogeneous* random field.

For example, we can model terrain as a random height field $h(x, y)$. A three-dimensional example would be the density map described earlier for clouds. A dynamic phenomenon such as cloud formation can be modelled by a four-dimensional random field $d(x, y, z, t)$. The function R can itself be of higher dimension. For example, if the function R models wind velocities then the random field R itself is three-dimensional.

2.2 Correlation Measures

The values of a random field are *independent* random variables if the value of one is unaffected by the value of the other. The *white noise* produced by such a field is unstructured, and on its own is not a useful model. A richer structure can be imposed by a *correlation measure* of the random field. Intuitively, a correlation measure defines how the values of the random field R at two given positions $\mathbf{t}$ and $\mathbf{s}$ are related. The most "natural" measure is the *variogram*, which is the mean-square difference of the random field at locations $\mathbf{t}$ and $\mathbf{s}$:

$$\gamma(\mathbf{t}, \mathbf{s}) = \frac{1}{2} E[(R(\mathbf{t}) - R(\mathbf{s}))^2]. \tag{3}$$

Another correlation measure is the *covariance*:

$$C(\mathbf{t}, \mathbf{s}) = E[R(\mathbf{t})R(\mathbf{s})] - \mu(\mathbf{t})\mu(\mathbf{s}). \tag{4}$$

Positive values of the covariance function indicate that the values of the random field at the two positions tend to be close. Conversely, negative values of the covariance indicate a probable large difference between the two values. The normalized version of the covariance is the *correlation* function:

$$\rho(\mathbf{t}, \mathbf{s}) = \frac{C(\mathbf{t}, \mathbf{s})}{\sigma(\mathbf{t})\sigma(\mathbf{s})}. \tag{5}$$

The variogram and the covariance are *second-order* statistics. We shall assume as others have that the second-order statistics are sufficient to characterize the visual characteristics of the phenomenon [8], [11]. If the phenomenon is purely Gaussian (i.e., $F_{\mathbf{t}}$ is a Gaussian distribution), the second-order statistics exactly specify the distribution.

An *isotropic* random field is one in which the correlation measure only depends on the distance between the two points $\mathbf{t}$ and $\mathbf{s}$. The covariance of an isotropic random field, for example, conveniently becomes a function of a single variable τ:

$$C(\mathbf{t}, \mathbf{s}) = C(\|\mathbf{t} - \mathbf{s}\|) = C(\tau). \tag{6}$$

In other words, for a given $\mathbf{t}$, the covariance is constant for all $\mathbf{s}$ lying on a sphere with centre $\mathbf{t}$ and given radius. The variogram and the correlation function of an isotropic random field are similarly univariate. In this case, the variogram can be obtained from the covariance by the following relation:

$$\gamma(\tau) = \sigma^2 + \mu^2 - C(\tau), \tag{7}$$

if C exists. The existence of the variogram does not guarantee that the covariance is well defined. A well-known counter-example is *Brownian motion* [22], a random process with undefined covariance, but with a variogram directly proportional to τ.

Isotropy is often a convenient property for reasons of computation and modelling. However, most natural phenomema have correlations with preferred directions. For example, the ripples on the surface of the sea are influenced by wind direction. Rather than immediately jump to nonisotropic random fields, however, we can generalize spherical correlations to *ellipsoidal* correlations [22]. For isotropic random fields, all points lying on the same sphere centred at a point $\mathbf{t}$ have the same correlation with $\mathbf{t}$. We can instead insist that all points lying on an ellipsoid about the point $\mathbf{t}$ have the same correlation. An ellipsoid is simply a scaling in the coordinate system of the random field. The covariance (or variogram) is now of the form:

$$C(\mathbf{s}, \mathbf{t}) = C((\mathbf{s} - \mathbf{t})\mathbf{Q}(\mathbf{s} - \mathbf{t})^t) \tag{8}$$

where $\mathbf{Q}$ is a $d \times d$ positive-definite and symmetric matrix, d being the dimension of the field. Setting $\mathbf{Q}$ to the identity matrix brings us back to a standard isotropic correlation. All the properties of the isotropic case are preserved in this more general setting.

Rather than specifying the matrix $\mathbf{Q}$ directly, we could instead specify the major axes and the eccentricity ϵ_i of the ellipsoid along each of these axes. From these values $\mathbf{Q}$ can be calculated automatically. We form a diagonal matrix $\mathbf{D}$ with respect to the coordinate system defined by these axes, with elements λ_i given by:

$$\lambda_i = \frac{1}{\epsilon_i^2}. \tag{9}$$

If $\mathbf{P}$ is the transformation matrix from the canonical coordinate system to the system given by the major axes of the ellipsoid, then

$$\mathbf{Q} = \mathbf{PDP}^t. \tag{10}$$

All covariance functions must possess the *positive definiteness* property, namely that for all points $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$ and coefficients $\lambda_1, \lambda_2, \ldots, \lambda_n$ we have the following inequality:

$$\sum_{i,j=1}^{n} \lambda_i \lambda_j C(\mathbf{t}_i, \mathbf{t}_j) \geq 0. \tag{11}$$

The inequality holds similarly for the correlation function. For isotropic random fields stronger conditions exist [22] [11].

2.3 Spectral Representation

Another way to characterize a random field is to analyse its frequency response using a Fourier transform. Let $\mathcal{R}(\boldsymbol{\omega})$ be the Fourier transform of the random field $R(\mathbf{t})$. As in [23] we define the *spectral density* function $S(\boldsymbol{\omega})$ as:

$$\lim_{T\to\infty} S_T(\boldsymbol{\omega}) = \lim_{T\to\infty} \frac{1}{T} |\mathcal{R}(\boldsymbol{\omega})|^2. \tag{12}$$

The Wiener-Khintchine theorem [22] states that for a homogeneous random field, the spectral density function and the covariance form a Fourier transform pair. Thus in theory these two functions have exactly the same modelling power. Since the Fourier transform preserves isotropy, the spectral density function of an isotropic random field is also isotropic. One simple way to generate a random field is to convolve a canonical random field such as white noise $W(\mathbf{t})$ with a deterministic filter kernel $H(\mathbf{t})$:

$$R(\mathbf{t}) = \int H(\mathbf{s} - \mathbf{t}) W(\mathbf{s})\, d\mathbf{s}. \tag{13}$$

In frequency domain, this reduces to multiplication:

$$\mathcal{R}(\boldsymbol{\omega}) = \mathcal{H}(\boldsymbol{\omega}) \mathcal{W}(\boldsymbol{\omega}). \tag{14}$$

If $S_W(\boldsymbol{\omega})$ is the spectral density of the random field $W(\mathbf{t})$, then the spectral density of the transformed random field $R(\mathbf{t})$ is [22]:

$$S_R(\boldsymbol{\omega}) = |\mathcal{H}(\boldsymbol{\omega})|^2 S_W(\boldsymbol{\omega}). \tag{15}$$

In the case of white noise, which has constant spectral density, the above equation gives us a direct way to construct $\mathcal{H}(\boldsymbol{\omega})$ from the desired spectral density function.

2.4 Nondeterministic fractals

Fractals can describe highly irregular phenomena, and they exhibit detail at all scales [13]. Some fractals are deterministically *self similar*. Exact self similarity is nonexistent in nature, but we can require that the random field modelling a phenomenon has second-order statistics that are self similar. The nondeterministic fractal *fractional Brownian motion* (fBm) has the statistical self-similarity property, and is characterized by the following variogram:

$$\gamma(\tau) \propto \tau^{2H}, \tag{16}$$

where $H = d + 1 - D$ is directly related to the fractal dimension D. $H = \frac{1}{2}$ gives Brownian motion. For any value of H, the random field has an infinite variance and hence the covariance is undefined. Because fBm is non-homogeneous, the Wiener-Khintchine theorem is inapplicable. The spectral density function for fBm can be derived heuristically, however, and is in fact [23]:

$$S(\boldsymbol{\omega}) \propto \|\boldsymbol{\omega}\|^{-\beta} \tag{17}$$

where $\beta = 2H + 1$. The spectral distribution is non-zero for all frequencies. This implies that fBm has detail at all scales. The statistical self similarity (or self affinity, [23]) is given by the relation

$$\gamma(a\tau) \propto a^{2H} \gamma(\tau), \tag{18}$$

where $a \geq 0$ is any scaling factor.

3 Previous Work

There have been many successful approaches to the modelling of natural phenomena using stochastic techniques. The main difficulties with most of these approaches for 3-D objects are high costs for storage and computation and a lack of control over the global shape of the phenomenon.

Spectral Models. Voss [23] was the first to suggest spectral models to generate random fields to simulate visual phenomena. By exploiting Eqs. 14,15,17, a frequency-domain characterization, $\mathcal{R}(\boldsymbol{\omega})$, of the phenomenon is created by filtering white noise. The random field $R(\mathbf{t})$ is just the inverse Fourier transform. The method is reasonably efficient for 2-D synthesis if the Fast Fourier Transform (FFT) is used. For 3-D phenomena, this technique suffers from the above problems. Particularly evident is the lack of control over the global shape of the phenomenon, which is an unpredictable trial-and-error process. Anjyo has recently generalized Voss work [1].

Stochastic Displacement. Fournier, Fussell and Carpenter introduced in [6] the most popular fractal based model: random midpoint displacement. The model is efficient and global shape can be controlled by specifying the value of the phenomenon at certain given points. Hence they call their algorithm *stochastic interpolation*. The algorithm recursively adds detail (higher frequencies), with new values being linearly interpolated from the old ones and then perturbed by Gaussian noise having zero mean and a variance satisfying Eq. 16. A lot of effort has been put into raytracing these models [5]. Again this method is memory-intensive and is hence unsuited for three-dimensional phenomena.

Constrained Fractals. Szeliski and Terzopoulos recently presented a new model to generate fractals [21]. The main advantage of their model is the possibility of controlling the global shape of the phenomenon. The model has two components, one smooth component, which is a spline approximating the data constraints provided by the user, and a stochastic component giving the fractal statistics. Interestingly, this model thus combines two popular modelling techniques in computer graphics into one. The model is generated by solving a variational problem. The quantity to be minimized is the "spline energy" and the "data constraint" energy. It turns out that the frequency response of the spline energy has a fractal spectrum (see Equation 17). As with the other techniques, it does not gracefully extend to 3-D phenomena.

Generalized Stochastic Subdivision. Lewis in [11] generalized the midpoint displacement algorithm to non-fractal random fields. He was the first to suggest the use of the correlation function as a modelling tool in computer graphics. His model is also procedural, requiring the solution of a linear system for the generation of each new value, as it is *estimated* from the previously-generated values. The technique he uses (*Wiener interpolation*) is similar to the estimation scheme described later in this paper. However, estimation in our model is used for a different purpose, namely to estimate the global shape. His model has the same drawbacks as the midpoint displacement algorithm, although it is not restricted to fractals.

Textured Ellipsoids. An algorithm similar in spirit to the one presented in this paper is presented by Gardner [7]. His model works essentially for density maps, which includes clouds and trees. Gardner uses the ellipsoid as the basic building block of his model. The user specifies the global shape of the phenomenon by arranging a set of ellipsoids. Small-scale detail is then added by using a (solid) texture. Gardner uses an analytical random function texture. Rendering is very simple: the translucence threshold is modified as a function of the projected equation of the ellipsoid onto the viewing plane. This threshold is high near the border of the ellipsoid and low near the centre of the ellipsoid.

Hypertexture. In the SIGGRAPH 89 proceedings we can find two 3-D modelling techniques that are similar in spirit [10] [17]. In both techniques the global shape of a phenomenon is defined using standard graphics primitives. Small-scale detail is then added by mapping a "thick" texture onto the global shape. Rendering is accomplished by rather expensive volume-rendering techniques. Both approaches give impressive results.

4 Overview of the Model

We will now present a new model for simulating visually a certain class of natural phenomena. As stated in the introduction we want an analytical model that permits a strong degree of control over the global shape, and over the small-scale random perturbation of the object. The perturbation is given by a random function, which is used as a solid texture [15] [16], although a variety of noise-synthesis techniques can be employed.

Our approach distinguishes between large scale and small scale visual detail. In our model, the user specifies: the value of the phenomenon at some arbitrary locations of space, and a correlation function describing how the values at these points are related. The global shape is smoothly interpolated from this data using *linear estimation*.

Small scale detail, which is produced by an analytic random function, makes the phenomenon "look real". Without it the object can appear too smooth and artificial. The user has control over this small scale by specifying the correlation function of the random field. We will describe below the classes of random functions suitable for generating small-scale detail. The advantages of choosing analytical random functions over random data bases (such as those generated by FFT based methods) are manifold: storage requirements do not increase exponentially with the dimension of the random field, and each value of the random field can be computed independently, hence the algorithm can be parallelized in a straightforward manner.

The model at both levels of scale uses a correlation measure. Unlike fractals, the correlation measures need not be the same. The model can also be stratified into more discrete levels of scale or generalized to continuous scale space.

5 Smooth Estimation

A user constrains the global shape by providing n pairs of data $(\mathbf{t}_i, d_i)$, where $\mathbf{t}_i$ is the location of the value d_i. The obvious way to get the global shape is by *smooth interpolation.* In smooth interpolation we look for a smooth function $L(\mathbf{t})$ such that

$$L(\mathbf{t}_i) = d_i \tag{19}$$

for $i = 1 \ldots n$. Furthermore we require that the function is "well behaved" away from the data locations, which precludes the use of Lagrange interpolation. A better choice would be thin-plate interpolation [21]. A more general solution is obtained if we view the interpolation problem as an *estimation* problem. In estimation theory, we wish to estimate the value of a random field at a certain location, given the knowledge of its values at a set of locations and its second order statistics. A popular estimation method first developed in geostatistics is called *Kriging* [8]. Kriging is a minimum variance, unbiased, linear estimation method which solves the following problem: given a random field $R(\mathbf{t})$ with known correlation $C(\tau)$ and a set of known values

$$d_1 = R(\mathbf{t}_1), d_2 = R(\mathbf{t}_2), \ldots, d_n = R(\mathbf{t}_n) \tag{20}$$

find a linear estimator

$$L(\mathbf{t}) = \sum_{i=1}^{n} \lambda_i d_i \tag{21}$$

such that $E[(L(\mathbf{t}) - R(\mathbf{t}))^2]$ is a minimum over all such (linear) estimators. To ensure uniqueness, we also require that the estimator be unbiased:

$$E[L(\mathbf{t})] = E[R(\mathbf{t})] = \mu. \tag{22}$$

This condition implicitly assumes that the random field is homogeneous. Later some extensions for non-homogeneous random fields will be mentioned. The above problem is a classical variational problem and can be solved by introducing a "Lagrange Multiplier" ν. The result is a linear system for the coefficients λ_i and the multiplier ν:

$$\boldsymbol{\lambda}\mathbf{M} = \mathbf{b}(\mathbf{t}) \tag{23}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n, \nu)$. The matrix $\mathbf{M}$ only depends on the covariance of the random field and on the location of the data points. More precisely, it is

$$\begin{pmatrix} C(0) & C_{12} & \cdots & C_{1n} & 1 \\ C_{21} & C(0) & \cdots & C_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{n1} & C_{n2} & \cdots & C(0) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}, \tag{24}$$

where C_{ij} stands for $C(\|\mathbf{t}_i - \mathbf{t}_j\|)$. The righthand side of Eq. 23 is a vector depending on the data locations and $\mathbf{t}$ only:

$$\mathbf{b}(\mathbf{t}) = (C(\|\mathbf{t} - \mathbf{t}_1\|), \ldots, C(\|\mathbf{t} - \mathbf{t}_n\|), 1). \tag{25}$$

Because $\mathbf{M}$ is positive-definite and symmetric (cf. Eq. 11), not only does Eq. 23 always have a solution, there are stable methods to compute it.

It may appear that the method is inefficient, because we have to solve a linear system for each location $\mathbf{t}$. However, using the symmetry of $\mathbf{M}$ and some linear algebra, we can prove that

$$L(\mathbf{t}) = \mathbf{b}(\mathbf{t})\mathbf{y}^t, \tag{26}$$

where $\mathbf{y}$ is the solution of the linear system

$$\mathbf{y}\mathbf{M} = \mathbf{d} \tag{27}$$

with $\mathbf{d} = (d_1, \ldots, d_n, 0)$. As $\mathbf{M}$ and $\mathbf{d}$ do not depend on $\mathbf{t}$, this system only has to be solved at most once per frame. Therefore our estimator is:

$$L(\mathbf{t}) = \sum_{i=1}^{n} y_i C(\|\mathbf{t} - \mathbf{t}_i\|). \tag{28}$$

If we consider a *Gaussian* covariance function $C(\tau) = \exp(-\alpha\tau^2)$ then L is a "blobby" [4].

In the case that the covariance is undefined, it is still possible to derive the above equations in terms of the variogram. In practice, however, we define a *pseudo-covariance* [8]:

$$C'(\tau) = A - \gamma(\tau), \tag{29}$$

and then apply the above equations to the pseudo-covariance instead of to the covariance. If the covariance exists, then A is actually equal to $\sigma^2 + \mu^2$ in accordance with Eq. 7. The quantity A is the asymptotic value of the variogram as τ tends to infinity. This assumes that the correlation structure is inherently local. For the fractal models this is not true in theory, but in practice it works as an approximation.

The non-bias condition of Eq. 22 assumes that the random field has the same mean for all locations $\mathbf{t}$. If the mean does depend on $\mathbf{t}$, i.e. $\mu = \mu(\mathbf{t})$, we introduce a homogeneous random field, called the *residual*,

$$S(\mathbf{t}) = R(\mathbf{t}) - \mu(\mathbf{t}). \tag{30}$$

It is now possible to apply the above Kriging procedure to the residual. However, the mean $\mu(\mathbf{t})$ may not be known at all. In this case, it is often assumed [8] that the mean has a simple form

$$\mu(\mathbf{t}) = \sum_{i=1}^{k} a_i f_i(\mathbf{t}) \tag{31}$$

where the coefficients a_i become additional unknowns of the Kriging equations and the f_i are a set of "basis functions". Usually the functions f_i are polynomials of a small degree.

6 Random Functions

We shall model the small-scale features with a simple random function. As with the large-scale level, this function is characterized by its second-order statistics. By a *simple* function we mean one that is given by a small number of coefficients and that can be evaluated at a point with no dependence on previous computations. We will now discuss two possibilities.

6.1 Spectral Sums

By Fourier analysis, we know that each random field can be approximated by a sum of spectral components. This is the basic idea behind the different "spectral sum" random functions. In [16] Perlin introduced a *noise function* $N(\mathbf{t})$. This function smoothly interpolates an integer lattice of independent Gaussian random variables. It is thus clearly bandlimited. Perlin uses this function as a basis for building more complicated random functions. By summing scaled versions of this function he is able to generate a fractal "$1/f$-noise":

$$\sum_i \left(\frac{1}{2}\right)^i N(2^i\mathbf{t}). \tag{32}$$

This corresponds to a fBm with spectral parameter $\beta = 3$, as shown in [19]. Mandelbrot in [13] gave an analytical random function, which is a modification of Weierstrass famous "nowhere differentiable but continuous" function:

$$W(t) = \sum_{k=-N}^{N} A_k r^{kH} \psi(\alpha r^{-k} t + \phi_k) \tag{33}$$

where the A_k are Gaussian random variables, ψ is any periodic function of period α, the ϕ_k are uniformly distributed random variables over the interval $[0,\alpha]$, H is the fractal codimension and r is the *lacunarity* factor. A thorough study of this function can be found in [3]. Several extensions to higher dimensions have been studied. Ausloos and Berman in [2] consider the following extension:

$$\Omega(\mathbf{t}) = \sum_{j=1}^{M} \omega_j W_j(\mathbf{n}_j \cdot \mathbf{t}) \tag{34}$$

Where the ω_j are uniformly distributed random variables, the W_j are one-dimensional Weierstrass-Mandelbrot functions and the $\mathbf{n}_j$ are unit vectors uniformly distributed over the unit sphere. The constants N and M are a tradeoff between image quality and efficiency. If M is too small, then directional artifacts become visible [20]. The Weierstrass-Mandelbrot function is interesting if we are interested in fractal random functions. All the above functions have a random component, unlike for example, Gardner in [7], who used a deterministic spectral sum. The apparent randomness is achieved by coupling the phases of the sinusoidal functions in the several coordinates.

6.2 Sparse Convolution

In [12] Lewis thoroughly reviewed most random functions and introduces a new random function to computer graphics:

$$P(\mathbf{t}) = \sum_i a_i h(\mathbf{t} - \mathbf{t}_i). \qquad (35)$$

This corresponds to a discrete version of a convolution, with kernel h, of a Poisson noise process. A Poisson noise process is an ensemble of uncorrelated values a_i, distributed at uncorrelated locations in space $\mathbf{t}_i$. The kernel entirely specifies the second-order statistics of the random function (see Eq. 15). The quality of the random function can be controlled by changing the density of the Poisson noise process.

7 Rendering of the Model

7.1 The Rendering Algorithm

While it is possible to render the above model using volume-rendering techniques, we prefer to use rendering techniques that exploit the geometry of the model and if necessary, develop special-purpose renderers to deal with common subcases of the model. A particularly desirable goal would be to develop techniques that can be incorporated into a standard scanline or ray-tracing renderer. Let us assume that the phenomenon to be rendered is a cloud. We will briefly outline the algorithm for a ray-tracing environment.

For each ray do
(1) Calculate intersection points of ray with
the isosurface of the global shape
(2) If no intersection then next ray
(3) Calculate brightness and self-shadowing at
the intersection point using the geometry of
the global shape
(4) Perturb brightness and calculate
translucency using the small-scale noise
function
(5) If translucence < 1 then continue to trace
the ray
Next ray

We will discuss step (1) in more detail in Section 7.2 and will present a simple algorithm for steps (3) and (4) based on Gardner's work [7] in Section 7.3.

7.2 Ray Tracing Generalized Blobbies

While it is possible to generalize the heuristic techniques used by Blinn [4] to ellipsoidal correlations, we prefer to use a more robust approach based on *interval arithmetic*, which was introduced to computer graphics by Mitchell [14]. Recall that our global model has the following form:

$$L(\mathbf{t}) - T = \sum_{i=1}^{n} y_i C((\mathbf{t} - \mathbf{t}_i)\mathbf{Q}_i(\mathbf{t} - \mathbf{t}_i)^t) - T, \qquad (36)$$

where T is a given threshold (defining the blobby as an isosurface), and the $\mathbf{Q}_i$ are as in Eq. 8, which gives an ellipsoidal shape to the correlation measure. When calculating the intersection of a ray and the blobby, we first transform the ray to its canonical form $R(t) = (0, 0, t)$, and then substitute $R(t)$ into $L(\mathbf{t})$ to get the equation

$$\varphi(t) = \sum_{i=1}^{n} y_i C(c_i + b_i(t - z_i) + a_i(t - z_i)^2) = T \qquad (37)$$

where a_i, b_i and c_i are given by the coefficients of $\mathbf{Q}_i$ and the components of $\mathbf{t}_i$. We isolate all the roots of φ in intervals using interval arithmetic. Once we have an interval $[a, b]$, which is guaranteed to contain only one root, we can apply a standard root finder such as the Newton iteration. For more details see [20].

7.3 The Illumination Model

From the global shape we get an intersection point P, the normal N to the surface and the distance d traversed through the cloud by the ray. First we calculate the classical illumination $I_{classic}$ given by any illumination model incorporating the ambient, diffuse and specular term. $I_{classic}$ accounts for semi-global illumination effects such as self-shadowing. We then texture the global shape by some random function R: $t = R(P)$, and compare this texture value t to a threshold T which is a function of the distance d and the cosine of the angle μ between the viewing vector and the normal N. We define a threshold T_1 at the "edge" of the cloud (typically very high) and a threshold T_2 at the "centre" of the cloud (typically very low). The threshold is varied in our current implementation as

$$T = T_1 + (T_2 - T_1)\exp(-\alpha d \mu^2) \qquad (38)$$

where α is a user specified parameter which influences the "fuzziness" of the cloud's "edge". The final texture value is set to

$$t_0 = \max(0, \nu(t, T)) \qquad (39)$$

where ν is a normalization function which is currently

$$\nu(t, T) = (t - T)/(1 - T). \qquad (40)$$

If I_{back} is the illumination coming from behind the cloud then the final illumination is given by:

$$I = t_0 I_{classic} + (1 - t_0)\exp(-\alpha d \mu^2) I_{back}. \qquad (41)$$

8 Results

As a test case, we have applied our model to clouds. Figures 1 and 2 depict sample data given by reddish spheres, with saturation of red denoting proximity to the viewer. These figures illustrate the global shape under an isotropic (spherical) correlation function with different parameters. Figures 3 and 4 depict the addition of various amounts of noise to the shape from Figure 3. Figure 5 gives a global shape resulting from the same data as in previous figures but using an ellipsoidal correlation function. Lastly, Figure 6 gives a resulting cloud after adding noise. Rendering costs are not high: on the order of 15-20 minutes for a 256×256 images on a SGI Personal Iris 25.

9 Future Work

We have introduced a new multiple-scale stochastic modelling primitive and have described a low-cost, low-storage rendering technique that can be embedded in a standard ray tracer. The directions to go from here are varied. The model permits great flexibility in the choice of correlation functions for the global shape. So far we have only used a Gaussian, but others are possible. We also wish to apply other noise-synthesis techniques to the local-scale model. Lastly, because our model is inherently geometric at the global scale, we have analytic values for depth and density through the object. This means that it may be possible to construct more realistic models containing terms for self-shadowing and refraction. The model is certainly not specific to clouds, and we hope to demonstrate other phenomena that are equally-well modelled.

10 Acknowledgements

We are grateful to Demetri Terzopoulos for inspiring some of this work and for pointing out the Kriging technique. We gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada, the Ontario Information Technology Research Centre, and the University of Toronto.

References

[1] K. A. Anjyo. "A Simple Spectral Approach to Stochastic Modelling for Natural Objects", *Proceedings of EUROGRAPHICS '88* (Sept. 1988), North-Holland, 285-296.

[2] M. Ausloos, D. H. Berman. "A Multivariate Weierstrass-Mandelbrot Function", *Proceedings of the Royal Society of London A 400* (1985), 331-350.

[3] M. V. Berry, Z. V. Lewis. "On the Weierstrass-Mandelbrot fractal function", *Proceedings of the Royal Society of London A 370* (1980), 459-484.

[4] J. F. Blinn. "A Generalization of Algebraic Surface Drawing", *ACM Transactions on Graphics, 1*, 3 (July 1982), 235-256.

[5] C. Bouville. "Bounding Ellipsoids for Ray-Fractal Intersection", *Proceedings of ACM SIGGRAPH '85*, also published as *ACM Computer Graphics 19*, 3 (July 1985), 45-52.

[6] A. Fournier, D. Fussell, L. Carpenter. "Computer Rendering of Stochastic Models". *Communications of the ACM, 25*, 6 (June 1982), 371-384.

[7] G. Y. Gardner. "Visual Simulation of Clouds", *Proceedings of ACM SIGGRAPH '85*, also published as *ACM Computer Graphics, 19*, 3 (July 1985), 297-303.

[8] A. G. Journel, C. J. Huijbregts. *Mining Geostatistics*, Academic Press, New York, 1978.

[9] J. T. Kajiya, B. P. Von Herzen. "Ray Tracing Volume Densities", *Proceedings of ACM SIGGRAPH '84*, also published as *ACM Computer Graphics 18*, 3 (July 1984), 165-174.

[10] J. T. Kajiya. "Rendering Fur with Three-Dimensional Textures", *Proceedings of ACM SIGGRAPH '83*, also published as *ACM Computer Graphics 23*, 3 (July 1989), 271-280.

[11] J. P. Lewis. "Generalized Stochastic Subdivision", *ACM Transactions on Graphics 6*, 3 (July 1987), 167-190.

[12] J. P. Lewis. "Algorithms for Solid Noise Synthesis", *Proceedings of ACM SIGGRAPH '89*, also published as *ACM Computer Graphics 23*, 3 (July 1989), 263-270.

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

[14] D. E. Mitchell. "Robust Ray Intersection with Interval Arithmetic", *Proceedings of Graphics Interface '90* (June 1990), 68-74.

[15] D. R. Peachey. "Solid Texturing of Complex Surfaces", *Proceedings of ACM SIGGRAPH '85*, also published as *ACM Computer Graphics 19*, 3 (July 1985), 279-286.

[16] K. Perlin. "An Image Synthesizer", *Proceedings of ACM SIGGRAPH '85*, also published as *ACM Computer Graphics 19*, 3 (July 1985), 287-296.

[17] K. Perlin, E. M. Hoffert. "Hypertexture", *Proceedings of ACM SIGGRAPH '89*, also published as *ACM Computer Graphics 23*, 3 (July 1989), 253-262.

[18] W. T. Reeves. "Particle Systems. A Technique for Modelling a Class of Fuzzy Objects", *Proceedings of ACM SIGGRAPH '83*, also published as *ACM Computer Graphics 17*, 3 (July 1983), 359-376.

[19] D. Saupe. "Point Evaluation of Multi-Variable Random Fractals", in *Visualisierung in Mathematik und Naturwissenschaft.* Springer-Verlag, Heidelberg, 1989.

[20] J. Stam. *A Multi-Scale Stochastic Model for Computer Graphics*, Master's thesis, Department of Computer Science, University of Toronto, 1991.

[21] R. Szeliski, D. Terzopoulos. "From Splines to Fractals", *Proceedings of ACM SIGGRAPH '89*, also published as *ACM Computer Graphics 23*, 3 (July 1989), 51-60.

[22] E. Vanmarcke. *Random Fields*, MIT Press, Cambridge, Massachussetts, 1983.

[23] R. F. Voss. "Fractals in nature: From characterization to simulation", in *The Science of Fractal Images*, Springer-Verlag, New York Berlin Heidelberg, 1988, 21-70.