A Multi-Scale Stochastic Model for Computer Graphics

Jos Stam

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

A Thesis submitted in conformity with the requirements
for the Degree of Master of Science in the
University of Toronto
o © Jos Stam 1991

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. The main contribution of this thesis is the introduction of a general stochastic modelling primitive that operates on two or more scales of visual detail. 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. A technique called *Kriging* is used to perform this interpolation. The microscopic level permits the addition of noise, which allows one to add interesting visual textural detail and translucency. A wide variety of noise-synthesis techniques can be employed in our model. The main advantages of the model over existing ones are low storage requirements and the use of geometric primitives that are amenable to rendering in traditional environments.

The basic theory of random fields, which underlies our model, and previous related models in computer graphics are thoroughly reviewed. As a case study we apply our model to the simulation of clouds. The rendering algorithm developed takes full advantage of the separation of scales inherent in our model. Two alternative rendering algorithms will be described to render clouds. Possible extensions to more scales and applications to other phenomena are also discussed.

Acknowledgements

Now that the dust has settled and the punctuation has been corrected, I want to thank a few people. Then I'll clean my apartment and eat a really balanced meal for the first time in a year and a half.

My thanks first to my supervisor Eugene for putting up with me at all stages of my thesis. He encouraged me when I doubted my work and helped me keep my sense of humour. My thanks also to Demetri whose "Visual Modelling" course reawakened my interest in stochastic modelling. Thanks also for pointing me to the Kriging technique and for being my second reader. Thanks to everyone in the graphics lab for providing two fundamental details: a good working environment and an unlimited supply of caffeine.

Thanks to all my new friends in Toronto without whom this thesis would have been finished suspiciously early.

Contents

List of Figures

Chapter 1

Introduction

The visual simulation of natural phenomena is an active research area in computer graphics. Unlike other applied sciences we are not interested in highly accurate models, which are essential for prediction and analysis. In computer graphics our goal is two-fold: (i) the model should capture the visual characteristics of the phenomenon, and (ii) it should permit control over the macroscopic features of the phenomenon. The first goal is part of the aim in computer graphics to produce "photo-realistic" images. Realism is important in applications such as flight simulators, and other applications in which a user should have the illusion of being in a natural environment. The second goal is essential in design and animation. In design, a user should be able to synthesize a particular form of the phenomenon he or she has in mind. In animation, control is important to achieve coherence between subsequent frames.

Many models have already been proposed in computer graphics which attempt to meet these criteria. In modelling three-dimensional fuzzy or partially-translucent phenomena, however, many approaches are hampered by high memory and computation requirements. Examples of such phenomena include clouds, fire, mist, smoke, dust and various types of solid (three-dimensional) textures.

The model presented in this thesis satisfies the two goals and overcomes the high memory requirements for the aforementioned class of phenomena. One of the main characteristics of our model is that it operates on two or more scales of visual detail. In order to grasp the complexities inherent in most natural phenomena, our model uses non-determinism in an essential way. A phenomenon is modelled as a random field whose second-order statistics may differ at different scales (unlike, for example, fractals).

In this thesis we will focus mainly on two-scale models. The first scale (macroscopic level) models the global shape of the phenomenon; it is constrained by a correlation measure that controls the interpolation of user supplied data values. We use a technique called *Kriging* which was first developed in mining geostatistics to perform this interpolation. The second scale (microscopic level) adds visual detail to the smooth global model. This scale is modelled by a random function with specified second-order statistics (correlation measure or spectral density function). The models at both scales have low storage requirements: for the global shape the interpolator is a weighted sum of basis functions, and for the small scale detail we only consider random function given by a small number of coefficients. For a certain class of correlation measures our global shape interpolator is a generalization of Blinn's "blobbies" [5].

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. Two possible rendering techniques for clouds are presented in detail. The first algorithm is a generalization of Gardner's method to render clouds [13]. It has the advantage of being efficient and implementable in a standard rendering software. The second approach uses volume-rendering techniques and achieves greater realism at the cost of higher computation time. Both techniques take advantage of the global shape information to simulate semi-global illumination effects, such as self-shadowing.

1.1 Thesis Overview

The thesis is organized as follows. In Chapter 2 we review the basic theory of random fields needed to understand the multi-scale model. We focus mainly on the second-order statistics of a random field, as they are the key to the whole modelling process. The theory of linear filters is briefly mentioned as it underlies most noise synthesis algorithms. At the end of the chapter we describe one of the most popular random fields used in computer graphics: *random fractals*.

In Chapter 3 most of the previously related models in computer graphics are presented and discussed. For each approach, the ability to model three-dimensional fuzzy or partially-translucent phenomena is discussed. Rendering issues are mentioned as well for each model.

In Chapter 4 we present our multi-scale stochastic model. We first give a general overview of the model, and then present the two scales separately. For the global shape, the theory of Kriging (which provides the interpolator) is reviewed in detail along with possible extensions. This is followed by a description of several existing random functions to model the small scale detail.

In Chapter 5 we present how our model can be applied to the simulation of clouds. Rendering issues will be addressed in detail. More precisely, two different rendering techniques will be presented: one simple heuristic technique which is a generalization of Gardner's work, and a more expensive algorithm based on volume rendering techniques.

In Chapter 6 we present several images synthesized using the model. The modelling process is presented first. The different options available to the user are then illustrated with concrete examples. Finally images of the rendering of three-dimensional clouds are discussed.

In Chapter 7 we summarize and comment on the contributions of this thesis and mention work in progress and possible directions of future research.

Chapter 2

Stochastic Modelling

2.1 Modelling in Computer Graphics

Computer graphics can be divided into two conceptual parts: modelling and rendering. In the modelling phase a mathematical model of the scene is generated from a description provided by a user. The rendering phase then takes this model as an input and generates a two-dimensional representation suitable for display on a raster device. In practice, however, both phases cannot be studied independently; of what use to computer graphics is a model that cannot be rendered? Conversely, a renderer that cannot process any interesting models is useless. Consequently, although this thesis is mainly concerned with the modelling of natural phenomena, the rendering of the model will also be discussed in detail.

2.1.1 Smooth Surfaces

Mainly for commercial reasons, most of the research in modelling has been devoted to geometric design. In the car industry, for example, computer graphics has revolutionized the design process. Instead of drawing or building a physical model of a car by hand, it is now possible to design substantial components of it using computer assisted systems. This makes the whole design process faster and provides more freedom to the designer. Most geometric design systems make use of *spline surfaces*, i.e. piecewise smooth surfaces. The whole design process is then a succession of stretching and bending of an initially flat piece of surface. In practice this is achieved by modifying the control points of the spline. Furthermore these surfaces can be displayed using standard renderers, e.g., by polygonalizing the surfaces. More generally, spline surfaces and related models turn out to be very successful at modelling 'man made' smooth objects, hence their success in industry. When dealing with natural phenomena, however, these models are often inappropriate. Nature exhibits great visual complexity, especially at small scales. Hence the aforementioned models are inappropriate for the simulation of natural phenomena. As an alternative the user could specify the entire phenomenon by an enormous number of polygons. This is the approach taken by Snyder and Barr in [36]. This approach is clearly not "user friendly" and is limited by storage capacity. Furthermore, it does not takes advantage of the possible regularities in the phenomenon.

2.1.2 Physically Based Modelling

Recently, a lot of research has been devoted to *physically based modelling* [39]. In this approach, the user does not specify the entire phenomenon, but rather provides external forces and material properties of the objects. The physical model then calculates, based on equations of classical mechanics, the position and shape of each object over time. This model clearly has the advantage of being inherently dynamical. Hence it is very well suited for certain animations. This approach, however, suffers from the limitations of classical mechanics. For some phenomena, although the equations exist and are well understood, physical simulation may be practically impossible. This is the essence of the so called *Science of Chaos* [14]. Due to the nonlinearities of the equations, two simulations having nearly the same initial condition can have a totally different behaviour. For example, two nearby smoke particles leaving a cigarette can end up at entirely different locations after some time. This clearly limits the control the user has over the phenomenon. For other complex natural phenomena, physical models do not yet exist and their study is still an area of research in physics or engineering.

2.1.3 Nondeterminism

What do we do in the case of unpredictable phenomena? We roll the dice! This is the approach adopted in most applied sciences when dealing with complex simulations. Instead of modelling the phenomenon deterministically, some amount of randomness is introduced to model the complexities of the model. This approach was introduced to computer graphics by Fournier, Fussell and Carpenter [12]. This approach is also called *data amplification*; the initial data which are being amplified are the macroscopic features and the statistical properties provided by a user. The model presented in this thesis falls into this category. The phenomenon is modelled as a random function defined in 3-space. It is therefore limited to phenomena which can be described as the set of values of a certain function. Examples of such phenomena include mountain terrain, hazy or partially translucent media and water. These types of functions are known as *random fields*. The basic theory of random fields will be reviewed in the next section. More precisely, first the notion of random variable will be explained, and then the second-order statistics of a random field will be introduced, specifically the variogram, covariance function, correlation function and the spectral density function. The famous Khintchin theorem is derived from simple heuristic arguments using the spectral properties of a random field. To understand most random field synthesis algorithms, the theory of linear filters will be briefly reviewed. At the end of this chapter we will be devoting a section to random fractals, because of its popular use in computer graphics.

2.2 Probability Theory Review

This section presents rather informally the basic notions of random fields. The goal is to emphasize the main concepts underlying the model, rather than giving a complete and rigorous mathematical presentation. For a rigorous presentation of the material see [43]. Most of the heuristic arguments are taken or adapted from [40].

2.2.1 Random Variables

A *random variable* X is a variable which takes all the possible values assigned to each realization of a phenomenon. For example, in the case where the phenomenon is "the tossing of a coin", then X can take only two values (e.g., 0 means "heads" and 1 means "tails"). The overall behaviour of a random variable (or sets of them) is governed by a *probability distribution*. In the case where the set of possible realizations is discrete, the probability distribution is simply a function $P(X = x)$ which assigns a probability (between 0 and 1) to the event x. Summing over all possible events gives:

$$\sum_{all\ possible\ x} P(X = x) \;=\; 1. \tag{2.1}$$

In the continuous case the probability distribution is a function f, where $f(x)$ is the probability that X has a value between x and $x + dx$. The continuous version of Equation 2.1 becomes:

$$\int_{all\ possible\ x} f(x)\,dx \;=\; 1. \tag{2.2}$$

A probability distribution thus entirely characterizes the stochastic phenomenon. In practice, this distribution is hard to find directly or does not exist. Thus, instead of working with the probability distribution we use *statistics* of the random variable X. A statistic is simply a function of the variable X that provides "useful" information about the phenomenon. The simplest statistic is the *expectation* of X:

$$\mu = E[X] = \int x\, f(x)\,dx. \tag{2.3}$$

Intuitively, μ is the mean value of the variable X. A measure of how the values of X are distributed around the mean is given by the *variance* statistic:

$$\sigma^2 = Var[X] = E[(X - \mu)^2] = E[X^2] - \mu^2. \tag{2.4}$$

The knowledge of these two statistics (μ and σ^2) entirely determines the probability density in case the phenomenon is *Gaussian*; more precisely, if the probability density function f is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{2.5}$$

This is one of the reasons why the Gaussian distribution is so popular in the applied sciences. For most other distributions one needs higher order statistics, more precisely statistics having terms

$$E[X^n] \tag{2.6}$$

where n is greater than 2. In many cases, however, these phenomena can be approximated by Gaussian distributions.

2.2.2 Random Fields

Most phenomena cannot be modelled using only a single random variable X. For example, if X is the value of the Dow Jones, then it would only model the behaviour of the stock market at a single instant of time. This is clearly not useful when one wants to predict future values! The solution is to have a different random variable at each time t. Hence

Figure 2.1: White noise

we define a *random process* as a function $X(t)$ which returns a random variable for each value of time t. More generally we can consider *random fields*, where we allow the domain of the function to have a higher dimensionality. For example we can model terrain as a height random field $h(x, y)$; in this case the argument is two-dimensional. An example of a three-dimensional domain can be given by clouds. Clouds can be modelled as density maps, for example when a value of one indicates total opacity and a value of zero stands for total translucence, the intermediate values give all the interesting visual effects. A dynamic phenomenon such as cloud formation can be modelled by a four-dimensional random field $d(x, y, z, t)$, for each fixed t we get an instance of the previous three-dimensional model. In the rest of this thesis a random field will be denoted by $R(\mathbf{t})$, where $\mathbf{t}$ is a variable of any of the above dimensions. Note that we can also allow the range of the function R to be of a higher dimension. For example, when wind velocities are modelled, the function R is three-dimensional. The behaviour of the random field is given by a probability density function which depends also on the argument $\mathbf{t}$. In other terms for each value of $\mathbf{t}$ a density function $f_{\mathbf{t}}(x)$ has to be specified. As a consequence, the two statistics introduced earlier now also become functions of the argument $\mathbf{t}$: $\mu = \mu(\mathbf{t})$ and $\sigma^2 = \sigma^2(\mathbf{t})$. In the next section we will introduce other statistics, which are specific to random fields.

2.2.3 Correlation Measures

Let us consider the most simple random field: for each value of $\mathbf{t}$ the function returns an *independent* random variable. Independence means that the value of the field at a certain location $\mathbf{t}$ is unaffected by the values at other points. What we get is something called *white noise*. Those who fall asleep in front of their television set are surely familiar with figure 2.1, however it doesn't model any interesting visual natural phenomena. Discernible structure is lacking. When we consider a mountain terrain, for example, we expect the heights at two

Figure 2.2: Two correlated random fields

nearby positions to be nearly the same (except of course near a steep cliff). This structure is captured by a *correlation measure*. Intuitively, the correlation measure tells us how the values of the random field R at two given positions $\mathbf{t}$ and $\mathbf{s}$ are related. Figure 2.2 shows two samples of a correlated random field. Clearly these pictures remind us of clouds. The most "natural" correlation measure is the *variogram*, which is basically the mean square difference of the random field at two locations $\mathbf{t}$ and $\mathbf{s}$:

$$\gamma(\mathbf{t},\mathbf{s}) = \frac{1}{2}E[(R(\mathbf{t}) - R(\mathbf{s}))^2]. \tag{2.7}$$

Another possible correlation measure is the *covariance*, which is defined by:

$$C(\mathbf{t},\mathbf{s}) = E[R(\mathbf{t})R(\mathbf{s})] - \mu(\mathbf{t})\mu(\mathbf{s}). \tag{2.8}$$

Intuitively, 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 in values. It is sometimes preferable to work with the normalized version of the covariance, which is the *correlation* function

$$\rho(\mathbf{t},\mathbf{s}) = \frac{C(\mathbf{t},\mathbf{s})}{\sigma(\mathbf{t})\sigma(\mathbf{s})}. \tag{2.9}$$

The functions just introduced along with the variance $\sigma^2(\mathbf{t})$ are the second-order statistics of the random field. In the rest of this thesis it will be assumed that second-order statistics are sufficient to characterize the phenomenon, the underlying assumption being that higher order statistics do not add more visual detail. This is widely assumed in most applied sciences; refer to computational vision [38] or geostatistics [18]. Lewis uses this assumption in computer graphics [23]. The domains of these functions have a dimensionality that is

twice as high as that of the random field. One way to lower this high dimensionality is to assume that the correlation measures only depend on the separation $\mathbf{h} = \mathbf{s} - \mathbf{t}$ between the two locations. Intuitively, this means that the phenomenon has similar statistics everywhere. The covariance function, for example, now becomes a function of the single variable $\mathbf{h}$:

$$C(\mathbf{t}, \mathbf{s}) = C(\mathbf{t}, \mathbf{t} + \mathbf{h}) = C(\mathbf{h}). \tag{2.10}$$

And the correlation function is equal to:

$$\rho(\mathbf{h}) = \frac{C(\mathbf{h})}{C(\mathbf{0})}, \tag{2.11}$$

meaning a simple scaling of the covariance. Another widely accepted assumption is to consider only random fields with constant means. That is, those $\mu(\mathbf{t})$ that do not depend on $\mathbf{t}$:

$$\mu(\mathbf{t}) = \mu \tag{2.12}$$

for all $\mathbf{t}$. Although there are many phenomena for which this assumption does not hold, it works well in practice as will be seen later in this thesis. The two above assumptions (dependence on distance only and constant mean) are referred to as the *homogeneity* assumption in the statistics literature [40] [43]. A random field satisfying these assumptions is called *homogeneous*. Without loss of generality we can assume this constant mean to be equal to zero. This simplifies the notations that follows.

Next we state some properties of these correlation measures which directly follow from their definitions and the above simplifying assumptions. The covariance and the correlation functions have the property of being *positive definite*. To see this consider n points $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$, and let Y be a random variable defined by the following linear combination:

$$Y = \lambda_1 R(\mathbf{t}_1) + \lambda_2 R(\mathbf{t}_2) + \cdots + \lambda_n R(\mathbf{t}_n). \tag{2.13}$$

From the fact that the variance of any random variable is positive, we get:

$$Var[Y] = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j C(\mathbf{t}_i - \mathbf{t}_j) \geq 0. \tag{2.14}$$

Dividing by $C(\mathbf{0}) > 0$ on both sides yields the same inequality for the correlation function (by Equation 2.11). Functions satisfying these inequalities for any choice of the coefficients λ_i are called *positive definite*, hence both the covariance and the correlation function fall into this class. These inequalities limit the choice of possible candidates for covariance and correlation functions. Conversely one can prove that any positive definite function is the covariance function of some random field [43], hence the covariance functions are exactly the positive definite functions. From the above inequalities we can deduce the following properties by appropriate choices of coefficients λ_i:

$$C(\mathbf{0}) > 0 \tag{2.15}$$

$$C(-\mathbf{h}) = C(\mathbf{h}) \tag{2.16}$$

$$|C(\mathbf{h})| \leq C(\mathbf{0}). \tag{2.17}$$

Similar relations can be obtained for the correlation function. The terms of Equation 2.14 can be arranged in a matrix, which is the *covariance matrix* of the random variables

$R(\mathbf{t}_1), R(\mathbf{t}_2), \ldots, R(\mathbf{t}_n)$:

$$\mathbf{C} = \begin{pmatrix} C(\mathbf{0}) & C(\mathbf{t}_1 - \mathbf{t}_2) & \cdots & C(\mathbf{t}_1 - \mathbf{t}_n) \\ C(\mathbf{t}_2 - \mathbf{t}_1) & C(\mathbf{0}) & \cdots & C(\mathbf{t}_2 - \mathbf{t}_n) \\ \vdots & \vdots & \ddots & \vdots \\ C(\mathbf{t}_n - \mathbf{t}_1) & C(\mathbf{t}_n - \mathbf{t}_2) & \cdots & C(\mathbf{0}) \end{pmatrix}. \tag{2.18}$$

The matrix is clearly symmetric, which means that all its eigenvalues are real, furthermore Equation 2.14 implies that these eigenvalues are positive. To see this take the λ_i to be equal to the components of an eigenvector corresponding to one of the eigenvalues. This fact will be used later in this thesis. The variance of the random variable Y can also be written down in terms of the variogram γ:

$$Var[Y] = C(\mathbf{0}) \sum_{i=1}^{n} \lambda_i \sum_{j=1}^{n} \lambda_j - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j \gamma(\mathbf{t}_i - \mathbf{t}_j) \geq 0. \tag{2.19}$$

If we assume that the sum of the coefficients vanishes, more precisely if

$$\sum_{i=1}^{n} \lambda_i = 0, \tag{2.20}$$

then we get an inequality very similar to Equation 2.14 for the variogram:

$$Var[Y] = -\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j \gamma(\mathbf{t}_i - \mathbf{t}_j) \geq 0. \tag{2.21}$$

Assumption 2.20 is very common in linear geostatistics [18], where one often passes from a formula for the covariance to a formula for the variogram simply by substituting $(-\gamma)$ for C. A consequence of Equation 2.21 [43] is that the variogram cannot grow faster than $\|\mathbf{h}\|^2$ as $\|\mathbf{h}\|$ tends to infinity. In fact,

$$\lim_{\|\mathbf{h}\| \to \infty} \frac{\gamma(\mathbf{h})}{\|\mathbf{h}\|^2} = 0. \tag{2.22}$$

Therefore, variograms can be unbounded, but their rate of divergence is bounded. The variogram also satisfies the following obvious properties:

$$\gamma(\mathbf{h}) \geq 0 \tag{2.23}$$

$$\gamma(\mathbf{h}) = \gamma(-\mathbf{h}) \tag{2.24}$$

$$\gamma(\mathbf{0}) = 0. \tag{2.25}$$

The covariance function and the variogram are in fact directly related, in certain cases. If we assume that the covariance exists, then a simple calculation shows that the variogram is given by

$$\gamma(\mathbf{h}) = C(\mathbf{0}) - C(\mathbf{h}). \tag{2.26}$$

This relation also directly implies that the variogram must be bounded by:

$$\gamma(\mathbf{h}) \leq 2C(\mathbf{0}), \tag{2.27}$$

and therefore a random field with an unbounded variogram cannot have a well-defined covariance function. The existence of the variogram in general does not guarantee that the covariance is defined. The best known counter-example is *Brownian motion*, whose variance is undefined (i.e., infinite), but has a well defined variogram that is known to be

$$\gamma(\mathbf{h}) \propto \|\mathbf{h}\|. \tag{2.28}$$

In some cases, however, when the random field has zero mean, $\mu(\mathbf{t}) = 0$, and the variogram tends asymptotically towards some value A, i.e.,

$$\lim_{\|\mathbf{h}\|\to\infty} \gamma(\mathbf{h}) = A, \tag{2.29}$$

then one can define a *pseudo-covariance* equal to

$$C'(\mathbf{h}) = A - \gamma(\mathbf{h}). \tag{2.30}$$

This definition coincides with Equation 2.26 when the covariance is well defined, and indeed $A = C(\mathbf{0})$.

2.2.4 Isotropic Random Fields

All the aforementioned correlation measures depend on a separation $\mathbf{h}$, which means that we can model phenomena with high anisotropies, i.e., with characteristics along preferred directions in space. A strong simplifying assumption is to consider instead *isotropic* random fields, isotropy meaning that the correlation measures depend only on the distance $\|\mathbf{h}\|$ between two points. In mathematical terms:

$$C(\mathbf{h}) = C(\|\mathbf{h}\|) = C(\tau). \tag{2.31}$$

Such correlation measures are now one-dimensional, and therefore we can restrict our analysis to one-dimensional functions, which highly simplifies the analysis involved. The class of isotropic correlation functions ρ is more restricted than the general case, as will be shown directly. All covariance functions have a lower bound inversely proportional to the dimension d of the domain of the random field. To see this, we consider $d+1$ points such that their relative distances all equal τ. Equation 2.14 with the λ_i all set to 1 is:

$$C(0)\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\rho(\tau) \geq 0. \tag{2.32}$$

As $\rho(0) = 1$ this expression becomes:

$$(d+1) + [(d+1)^2 - (d+1)]\rho(\tau) \geq 0 \tag{2.33}$$

or, after simplifications:

$$\rho(\tau) \geq -\frac{1}{d}. \tag{2.34}$$

Hence this gives a lower bound for an isotropic correlation function, the bound getting tighter with the dimension. In [43] tighter bounds are given, using properties of the spectral density function (this function will be defined in the next section). For example, for $d = 2$ the bound is -0.403 and for $d = 3$ it is -0.218.

Many natural phenomena are highly anisotropic. Consider for example the ripples on the surface of the sea caused by the wind. However these anisotropies can be modelled by transforming an isotropic correlation measure. For isotropic random fields, all points lying on the same sphere centred at a point $\mathbf{t}$ have the same correlation with $\mathbf{t}$. One can instead insist that all points on an ellipsoid about $\mathbf{t}$ have the same correlation. An ellipsoid can be obtained from a sphere by a simple scaling transformation along a set of axes. Hence we can define the covariance (or the variogram) as:

$$C(\mathbf{h}) = C(\mathbf{h}^t \mathbf{Q} \mathbf{h}) \tag{2.35}$$

where $\mathbf{Q}$ is a $d \times d$ positive-definite and symmetric matrix and d is the dimension of the domain of the random field. $\mathbf{h}^t$ denotes the transpose of the vector $\mathbf{h}$. Correlation measures defined in such a manner are called *ellipsoidal* [40]. 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. The matrix $\mathbf{Q}$ is a rather unintuitive way to specify anisotropies for a general user. The user could instead specify the major axes and the corresponding eccentricities e_i. 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:

$$\lambda_i = \frac{1}{e_i^2}. \tag{2.36}$$

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

$$\mathbf{Q} = \mathbf{P}^t \mathbf{D} \mathbf{P}. \tag{2.37}$$

2.2.5 Spectral Representation of a Random Field

In this section we will review the characteristics of a homogeneous random field in the frequency domain. The main result is that the spectral density function and the covariance form a Fourier transform pair. To show this result, the approach taken by VanMarcke in [40] will be used, which does not require the use of complex random fields. Furthermore, only the case where the domain is one-dimensional will be considered, since the extension to higher dimension is straightforward [40].

One of the most elementary homogeneous random fields is the random harmonic oscillator of frequency ω, defined by:

$$X(t) = A\cos(\omega t + \phi), \tag{2.38}$$

where A is a random variable with zero mean and ϕ is uniformly distributed over the interval $[0, 2\pi]$. Furthermore these two random variables must be independent. The harmonic oscillator has zero mean and a variance given by:

$$\sigma^2 = E[X^2(t)] = E[A^2]E[\cos^2(\omega t + \phi)] = \frac{1}{2}E[A^2]. \tag{2.39}$$

This shows that the variance is directly proportional to the average energy (or power) of the oscillator. By using some basic trigonometry, it is possible to calculate the covariance as well:

$$C(\tau) = E[X(\tau)X(0)] = \sigma^2 \cos(\omega\tau). \tag{2.40}$$

Let us now consider any homogeneous random field $R(t)$. It can be proven [43], that any such field can be closely approximated arbitrarily by a sum of (random) harmonic oscillators:

$$R(t) \approx \sum_{i=-N}^{N} R_i(t) \tag{2.41}$$

where

$$R_i(t) = A_i \cos(\omega_i t + \phi_i). \tag{2.42}$$

The random amplitudes A_i and random phase angles ϕ_i are mutually independent. Hence the variance of the random field $R(t)$ is given by

$$C(0) = \sigma^2 = \sum_{i=-N}^{N} \sigma_i^2 = \sum_{i=-N}^{N} \frac{1}{2} E[A_i^2]. \tag{2.43}$$

Let us assume that the *spectrum*, i.e. the set of frequencies ω_i, is a uniform partition of the interval $[-N, N]$. More specifically the frequencies are given by $\omega_i = \Delta\omega(2i-1)/2$. This analysis shows that the variance (power) is distributed over the discrete frequencies ω_i. This suggest the introduction of a spectral mass function:

$$S(\omega_i)\Delta\omega = \frac{1}{2} E[A_i^2]. \tag{2.44}$$

The function S is called the *spectral density function*, and gives the contribution of each frequency to the total variance (power) of the random field. Until now we have only considered discrete frequencies. The usual way to extend the theory to the continuous case (for applied mathematicians !) is to let N and $\Delta\omega$ tend to infinity and zero respectively, while holding their product constant. By taking these limits in Equation 2.43 and using Equation 2.44, we get:

$$C(0) = \int_{-\infty}^{+\infty} S(\omega)\, d\omega. \tag{2.45}$$

We now try to find a similar relation between the correlation function and the spectral density function. We begin by observing that

$$C(\tau) = E[R(\tau)R(0)] = \sum_{i=-N}^{N} C_i(\tau). \tag{2.46}$$

Knowing the covariances C_i of each oscillator, the covariance can be expressed in terms of the spectral density function:

$$C(\tau) = \sum_{i=-N}^{N} S(\omega_i)\Delta\omega \cos(\omega_i \tau). \tag{2.47}$$

Using the same limit argument as before we finally get the relation in the continuous case:

$$C(\tau) = \int_{-\infty}^{+\infty} S(\omega) \cos(\omega\tau)\, d\omega. \tag{2.48}$$

This is the famous theorem of Khintchin, which states that the spectral density function and the covariance form a Fourier transform pair. Therefore both functions have exactly the same modelling power from a theoretical point of view. The Khintchin theorem also

provides us with a characterization of the allowed covariance functions for homogeneous random fields, namely as the inverse Fourier transform of any positive function $S(\omega) \geq 0$. Because of Equation 2.26, the variogram is also related to the spectral density function by:

$$\gamma(\tau) = \int_{-\infty}^{+\infty} S(\omega)(1 - \cos(\omega\tau))\, d\omega. \tag{2.49}$$

In the case of isotropic random fields it is unnecessary to consider spectral densities with domains of dimension higher than one. This is because the Fourier transform preserves isotropy, hence the spectral density of a random field with isotropic covariance is also isotropic. However, not all inverse Fourier transforms of positive functions are covariance functions of isotropic random fields. One has to consider other transforms instead. If $S(\omega)$ is any positive function, then all the two-dimensional isotropic covariance functions are given by:

$$C(\tau) = 2\pi \int_{0}^{+\infty} S(\omega) J_0(\omega\tau)\omega\, d\omega \tag{2.50}$$

where J_0 is the zero order Bessel function of the first kind [43]. For the three-dimensional case, all such covariance functions are given by the transformation:

$$C(\tau) = 4\pi \int_{0}^{+\infty} S(\omega) \frac{\sin(\omega\tau)}{\omega\tau} \omega^2\, d\omega. \tag{2.51}$$

Similar transformations exist for higher dimensional isotropic random fields [43]. There is a surprising result mentioned in [40], which is only true for isotropic spectral density functions of random fields with a three-dimensional domain. The result is that for this case the spectral density function must be monotonically decreasing. In particular no oscillating functions are allowed. Note that this does not imply that the correlation function has to be monotonically decreasing, as this property is not necessarily preserved by the Fourier transform.

2.2.6 Transforming Random Fields

Until now we have not assessed the problem of actually generating a random field with prescribed correlation function (or spectral density function). One way to generate such a random field is to transform a random field that is easy to generate. In most cases the latter will be *white noise*, which has a correlation function that is the delta function and that has a constant spectral density function.

One of the most common transformations is the *shift invariant linear filter*, which is well known in signal theory. A filter can be viewed as a black box, which, when given an input signal $x(\mathbf{t})$ produces an output signal $y(\mathbf{t}) = L\{x(\mathbf{t})\}$. A linearity condition is imposed on the filter, i.e. if x_1 and x_2 are two signals and λ is a constant then

$$L\{\lambda x_1(\mathbf{t}) + x_2(\mathbf{t})\} = \lambda L\{x_1(\mathbf{t})\} + L\{x_2(\mathbf{t})\}. \tag{2.52}$$

The filter is *shift invariant* if the following equality holds for all input signals $x(\mathbf{t})$:

$$L\{x(\mathbf{t} + \mathbf{h})\} = y(\mathbf{t} + \mathbf{h}). \tag{2.53}$$

One such filter is explicitly given by:

$$y(\mathbf{t}) = \int k(\mathbf{s} - \mathbf{t}) x(\mathbf{s})\, d\mathbf{s}. \tag{2.54}$$

This is actually a *convolution* of the signal $x(\mathbf{t})$ with the *convolution kernel* $k(\mathbf{t})$. The filter is thus entirely specified by the function $k(\mathbf{t})$. As in the previous section we explore what happens in the "frequency domain". Let $X(\boldsymbol{\omega})$, $Y(\boldsymbol{\omega})$ and $K(\boldsymbol{\omega})$ denote the Fourier transforms of $x(\mathbf{t})$, $y(\mathbf{t})$ and $k(\mathbf{t})$ respectively. By applying the Fourier transform to both sides of the equality in Equation 2.54 and using some calculus we get

$$Y(\boldsymbol{\omega}) = K(\boldsymbol{\omega})X(\boldsymbol{\omega}). \tag{2.55}$$

It is the well known result that a convolution corresponds to a direct multiplication in the frequency domain. Now we understand why this transformation is called a filter. A well-chosen function K attenuates or amplifies the values of the input signal X. For example if K is a box centred at the origin, then it acts as a "low pass" filter, killing all the higher frequencies (typically unwanted noise) of the input signal X. Let us now return to random fields. Let the input signal $x(\mathbf{t})$ be a random field with known covariance and spectral density functions $C_x(\boldsymbol{\tau})$ $S_X(\boldsymbol{\omega})$ respectively. A simple calculation shows that the covariance $C_y(\boldsymbol{\tau})$ of the output random noise $y(\mathbf{t})$ is given by

$$C_y(\boldsymbol{\tau}) = \int\int k(\boldsymbol{\xi})k(\boldsymbol{\eta})C_x(\boldsymbol{\tau} + \boldsymbol{\xi} - \boldsymbol{\eta})\, d\boldsymbol{\xi} d\boldsymbol{\eta}. \tag{2.56}$$

Thus by choosing the right kernel k we can get the desired correlation. There exists a similar relation between the spectral density functions. This relation is established as before by taking the Fourier transform on both sides of the equality of Equation 2.56 and using some basic calculus:

$$S_Y(\boldsymbol{\omega}) = |K(\boldsymbol{\omega})|^2 S_X(\boldsymbol{\omega}). \tag{2.57}$$

Again the relation is simpler in the frequency domain. The Fourier transform of the kernel K is easier to construct than the kernel itself. To summarize, Equations 2.54 and 2.55 tell us how to generate the random field, and Equations 2.56 and 2.57 tell us how to find the kernel. At first glance, modelling in the frequency domain with the spectral density function seems much easier. The problem, however, is that we do not live in the world of frequencies and moreover it is not always easy to think in terms of them. After having generated $Y(\boldsymbol{\omega})$ we have to take an inverse Fourier transform to get the desired random field $y(\mathbf{t})$. As we will see in subsequent sections, this imposes certain practical limitations on the spectral approach.

To have a better understanding of the procedure described above let us consider a specific example. We shall consider the case where the input random field $x(\mathbf{t})$ is white noise; its second-order statistics are given by

$$C_x(\boldsymbol{\tau}) \propto \delta(\boldsymbol{\tau}) \tag{2.58}$$

$$S_X(\boldsymbol{\omega}) \propto 1 \tag{2.59}$$

where δ is the Dirac delta (generalized) function. Because of the "point sampling" property of the delta function:

$$\int k(\boldsymbol{\eta})\delta(\boldsymbol{\tau} - \boldsymbol{\eta})\, d\boldsymbol{\eta} = k(\boldsymbol{\tau}), \tag{2.60}$$

the second-order statistics of the output field $y(\mathbf{t})$ are given by:

$$C_y(\boldsymbol{\tau}) \propto \int k(\boldsymbol{\xi})k(\boldsymbol{\xi} + \boldsymbol{\tau})\, d\boldsymbol{\xi} \tag{2.61}$$

$$S_Y(\boldsymbol{\omega}) \propto |K(\boldsymbol{\omega})|^2. \tag{2.62}$$

In frequency domain the convolution kernel is thus directly given by the (desired) spectral density function of the output random field. The determination of the kernel from the covariance function is less straightforward, but is nevertheless possible in certain cases.

2.3 Random Fractals

2.3.1 What is a Fractal ?

In this section the theory of a certain class of random fields will be reviewed: *random fractals*. Random fractals were first studied by B. Mandelbrot in his now famous *Fractal Geometry of Nature* [25]. Fractals have been widely used in computer graphics, and constitute the most popular subclass of stochastic models [12]. Fractals are especially well suited to the description of highly irregular phenomena, exhibiting detail at all scales, the most striking example being the Mandelbrot Set. Most fractals, however, also have the *self-similarity* property, which roughly means that the fractal contains copies of itself at smaller scales. A good illustration of the latter is the *von Koch Snowflake*. Exact self-similarity (such as exhibited by the von Koch snowflake) is nonexistent in Nature. Therefore Mandelbrot introduces the concept of statistical self-similarity. This concept will be defined more precisely later. We will first consider the one-dimensional case, and then mention extensions to higher dimensions. There are many ways to define a random fractal. The usual approach [25] [26], is to define a random fractal as *fractional Brownian motion* (fBm). That is, a random field $F(t)$ whose variogram is given by:

$$\gamma(\tau) \propto |\tau|^{2H}. \tag{2.63}$$

Because of the constraint on the growth of the variogram, given by Equation 2.22, the parameter H cannot exceed 1, and by continuity we have $H > 0$. This parameter is directly related to the *fractal dimension* D [41] of the fractal, by the relation $H = 2 - D$. In particular, contrarily to our intuitive notion of (euclidean) dimension, D can take non-integer values. The fractal dimension lies actually somewhere between 1 ($H = 1$) and 2 ($H = 0$), which means intuitively that, when embedded in a plane, the fractal is an object between a straight line and a plane. Thus, highly irregular curves which tend to fill the plane have fractal dimensions close to 2, and curves that slightly deviate from a line tend to have dimensions close to 1. For $H = \frac{1}{2}$ we get ordinary Brownian motion. The variogram is self-similar in the sense that for any scaling factor $a > 0$ we have:

$$\gamma(a\tau) \propto a^{2H}\gamma(\tau). \tag{2.64}$$

Hence, no characteristic scale can be associated to the variogram. Furthermore, it can be proven that only variograms of the form defined by Equation 2.63 are self-similar in this sense.

2.3.2 Spectral Analysis of a Fractal

The variogram of a fractal is clearly unbounded for all the values of H considered, hence, according to Equation 2.27, an fBm has no defined covariance and cannot be homogeneous. Therefore all the theory previously developed cannot be applied directly to random fractals! Fortunately, the increments of an fBm are homogeneous, so the previous theory can be applied to these increments. It can be shown [43] that in fact Equation 2.49 remains valid.

Therefore, the spectral density function of the random fractal is given by the following equation:

$$\tau^{2H} \propto \int_{-\infty}^{+\infty} S(\omega)(1 - \cos(\omega\tau))\, d\omega. \tag{2.65}$$

Using integration by parts it is not to hard to verify that a spectral density function equal to

$$S(\omega) \propto \frac{1}{\omega^{\beta}} \tag{2.66}$$

is a solution of the above equation, if $\beta = 2H + 1$. The spectral density is non-zero for all frequencies. This implies that an fBm has detail at all scales, which with the self-similarity property, is the most important characteristic of fractals. The exponent β is related to the fractal dimension D by

$$D = \frac{5 - \beta}{2}. \tag{2.67}$$

As we decrease the exponent β (and hence increase the fractal dimension D), the spectral density function takes higher values at the high frequencies and smaller values at the low frequencies. This increase in high frequencies results in increased small-scale deviations, hence the curve will tend to be plane filling. An increase of the exponent β has the exact reverse effect: the low frequencies tend to dominate, and hence the curve has mainly large-scale deviations and is geometrically closer to a straight line.

The obvious way to generalize these definitions to higher dimensions [41], is to assume that the variogram is isotropic, namely that

$$\gamma(\mathbf{h}) \propto \|\mathbf{h}\|^{2H}. \tag{2.68}$$

The spectral density function then is equal to

$$S(\boldsymbol{\omega}) \propto \|\boldsymbol{\omega}\|^{-\beta}. \tag{2.69}$$

If the dimension of the argument is d then the parameters H and β are related to the fractal dimension D by the following relations:

$$H = d + 1 - D \tag{2.70}$$

$$\beta = 2d - 2D + 3. \tag{2.71}$$

Chapter 3

Previous Work

In this chapter we review previous models of natural phenomena. For each model we discuss its applicability to three-dimensional fuzzy and partially translucent phenomena. The main difficulty with most models is their high storage cost. Most of these models only synthesize random fractals. These are: the spectral approach of Voss, the stochastic displacement technique of Fournier et al. and more recently the constrained fractals of Szeliski and Terzopoulos. Lewis generalizes stochastic displacement to random function with arbitrary correlation functions. Gardner's model is specific to clouds. The models of Kajiya and of Perlin and Hoffert are alternatives to stochastic modelling for simulating natural phenomena.

3.1 Spectral Models

Spectral models are characterized by the fact that the phenomenon is entirely specified by the spectral density function. The most common approach is to filter white noise in the frequency domain, and then to take the inverse Fourier transform in order to obtain the resulting phenomenon. Voss [41] was the first to suggest a practical approach to simulate visual phenomena. He considers only fractal spectral density functions, namely those given by Equation 2.66. Let us consider in more detail how his algorithm works. The random field Y is first generated in the frequency domain by filtering a white noise $W(\boldsymbol{\omega})$, according to equation 2.55:

$$Y(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|^{-\frac{\beta}{2}} W(\boldsymbol{\omega}). \tag{3.1}$$

Next Y is sampled at a set of N discrete frequencies $\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{N-1}$. Then the random field $y(\mathbf{t})$ is generated at a set of discrete points $\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_{N-1}$ using the discrete Fourier transform (in practice, the Fast Fourier Transform):

$$y_k = y(\mathbf{t}_k) = \sum_{l=0}^{N-1} Y(\boldsymbol{\omega}_l) \exp(2\pi i \boldsymbol{\omega}_l \cdot \mathbf{t}_k) \tag{3.2}$$

where "$\cdot$" denotes the standard dot product of two vectors. This method is relatively fast (due to the $N \log N$ complexity of the FFT). The major drawback is that all the coefficients y_k have to be generated at the same time for the FFT algorithm to be usable. This is particularly prohibitive in terms of storage for the three-dimensional case (e.g. for clouds). As an example, if the sampling grid is $256 \times 256 \times 256$ we must store (at least) 2^{24} bytes; i.e., 16 megabytes! The major drawback, however, of this algorithm as a modelling tool, is

the lack of control over the global shape of the phenomenon. The only way to change the global shape is to modify the random seed of the white noise generator. The global shape one obtains in this way is, however, totally unpredictable. Control over the global shape is important in dynamical simulations and design. Furthermore no efficient algorithms have yet been developed to render volumes given as a set of values at discrete points, especially if the phenomenon in question is a density map, such as a cloud. Kajiya and VonHerzen have developed such physically based rendering algorithm for the case of density arrays [20], which is a brute force extension of Blinn's earlier work [6], but the computation times involved remain prohibitive. Voss, however, has produced some very impressive pictures of clouds by modelling a three-dimensional cloud as an ensemble of "thin" two-dimensional clouds. This cuts down the storage requirements and simplifies the rendering somewhat.

Recently Anjyo [1] has generalized Voss's work for spectral distributions of the form

$$S(\omega) \propto \frac{\omega^m}{(c + \omega^\beta)^n} \tag{3.3}$$

where m, c, β and n are parameters. For $m = c = 0$ and $n = 1$ we get the fractal model. The final random field $R(\mathbf{t})$ is given by a sum of N *stochastic waves* $W_i(\mathbf{t})$. Each stochastic wave is the product of two one-dimensional random process X_{i1} and X_{i2}, generated by the spectral method, along the directions $\mathbf{d}_{i1}$ and $\mathbf{d}_{i2}$:

$$W_i(\mathbf{t}) = X_{i1}(\mathbf{d}_{i1} \cdot \mathbf{t}) X_{i2}(\mathbf{d}_{i2} \cdot \mathbf{t}). \tag{3.4}$$

For example, it is possible to approximate the known spectral distribution of ocean waves with this model. His approach, although more general, suffers from the same limitations as Voss's model.

3.2 Stochastic Displacement

Fournier, Fussell and Carpenter introduce in [12] the most popular fractal based model; random midpoint displacement. Their model is very efficient: it can be implemented by using only addition and shift operations. Furthermore the global shape can be controlled by specifying the value of the phenomenon at certain given points. Hence they call their algorithm *stochastic interpolation*. As in Voss's model, it is limited to fBms. Instead of using the spectral characterization of fBm, they use the variogram as a modelling tool. The algorithm is recursive: at each step more detail (higher frequencies) is added by refining the sampling grid into twice as many samples in each direction. The new values are linearly interpolated from the old ones and then perturbed by some Gaussian noise having zero mean and a variance which must satisfy Equation 2.63. A lot of effort has been put into ray-tracing these models ([19] and [8]). Again this method requires a lot of memory and hence is unsuited for three-dimensional phenomena. Furthermore, as noted in [41], this model produces only true fBm in the case of Brownian motion ($H = \frac{1}{2}$), because only the new points are perturbed at each recursion level. This shortcoming may produce visible creases in the case of terrain modelling, though whether or not such artifacts are visually annoying is arguable. Voss overcomes this problem by updating all the points at each level [41].

3.3 Generalized Stochastic Subdivision

Lewis in [23] generalizes the midpoint displacement algorithm for non-fractal random fields. The statistics of the random field are specified by a correlation function. He was the first to suggest the use of the correlation function as a modelling tool in computer graphics. Instead of linearly interpolating the new values at each level, the values are *estimated* from a set of points at the coarser level. The estimation procedure involves the solution of a linear system, which depends on the correlation function. As the model presented in this thesis also uses an estimation procedure, we will delay the explanation of estimation methods to a later chapter. Lewis' model has the same drawbacks as the previous stochastic subdivision algorithm. Because fractals have no defined covariance, this model is unable to generate fBms. It therefore does not truly generalize the previous stochastic subdivision algorithm.

3.4 Constrained Fractals

Recently, Szeliski and Terzopoulos presented in [37] a new model to generate fractals. The main advantage of their model is the possibility of controlling the global shape of the phenomenon. The model has two components; a smooth component (a deterministic spline which approximates the data constraints provided by the user) and a stochastic component that gives the fractal statistics. This model thus synthesizes 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 sum of the "spline energy" and the "data constraint" energies. It turns out that the frequency response of the spline energy has a fractal spectrum (see Equation 2.66). As energies are related to probabilities by the Boltzmann equation of thermodynamics, the variational problem can be interpreted as a *maximum a posteriori* (MAP) estimation problem, where one wants to generate the sample with the highest probability, given the data constraints and the fractal *a priori model*. The variational problem is solved by using a stochastic coarse-to-fine relaxation scheme. At each relaxation step a certain amount of noise is added, where the variance is proportional to the grid level. This method, however, has the same storage requirement problem as the FFT-based approach and, hence, is unsuited for three-dimensional phenomena such as clouds.

3.5 Textured Ellipsoids

A model more in the spirit of the one presented in this thesis was proposed by Gardner [13]. 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 an ensemble of ellipsoids. The small-scale detail is then added by using a (solid) texture. Gardner uses an analytic random function texture. This texture function will be studied in more detail in a later chapter. His rendering algorithm is very simple. He modifies the translucence threshold as a function of the projected equation of the ellipsoid onto the viewing plane. This threshold is very high near the border of the ellipsoid and very low near the centre of the ellipsoid. His rendering algorithm can easily be implemented into a standard ray tracer. The drawback of this model is that it is somewhat restricted to ellipsoids and translucent phenomena.

3.6 Thick Textures

In the SIGGRAPH 89 proceedings one can find two models that are similar in spirit. In both models one defines the global shape of the phenomenon with standard graphics primitives, e.g. polygons or patches, and then adds small-scale detail by mapping a "thick" texture onto it. Kajiya [21] calls this texture *texels*. A texel is a cube containing *microsurfaces*. Instead of specifying all the microsurfaces, Kajiya introduces three functions for each texel: (i) a scalar density function, which basically measures the density of microsurfaces at a specific point of the cube; (ii) a frame bundle, which is a collection of three vectors specifying the local orientation of the microsurfaces within the cube; (iii) a bidirectional light reflection function that indicates the surface properties of the microsurface. The entire "thick" texture is given by mapping many texels onto the object. Of course one has to be careful that the texels blend together in a smooth way. Kajiya uses this model to simulate fur, and has generated the most impressive image of a teddy bear to date. However the computation times are far more than prohibitive.

The other model is due to Perlin and Hoffert [31]. In this model all phenomena are simulated as density functions $D(\mathbf{x})$. The *soft region* of the object is defined as the ensemble of $\mathbf{x}$ for which $0 < D(\mathbf{x}) < 1$. The modelling power comes from the transformation of the soft region by different functions. This extends Perlin's earlier work in *functional modelling* [30]. With this model Perlin and Hoffert are able to model fur, fire and glass. The rendering is accomplished by using expensive volume-rendering techniques.

Chapter 4

The Model

> The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.
> *John Von Neumann*

In this chapter we present the main contribution of this thesis, namely a new multi-scale stochastic model for computer graphics. In the next section we describe the importance of scales in modelling natural phenomena, and show how our model separates the scales. In the second section the model of the global shape is presented along with the theory of Kriging. The third section describes many existing random functions which can be used to model the small-scale detail. We conclude this chapter with a model called stochastic Kriging which combines the two scales elegantly, and we will mention possible extensions of the model to more scales.

4.1 Separation of Scales

When we observe a natural phenomenon, we find that its visual characteristics change with scale. Consider for example a mountain: at a very large scale (i.e., viewed from far away) only the major trends of the relief are visible, at a medium scale (i.e., by looking only at a portion of the mountain) we start to observe the distribution of vegetation and different rock types, at the small scale (i.e., when observing the ground) we see the structure of a particular vegetation or rock type. There is no reason for the correlation measures to be the same at each scale. All fractal models in computer graphics, however, assume that the statistics are the same at all scales, because of the statistical self-similarity property given by Equation 2.64. Therefore it seems appropriate to consider *multi-scale models* in which the final model is the sum of several models at different scales. The model at each scale is specified by a different correlation measure. The correlation measure alone, however, is insufficient. We also want to be able to constrain the values of the phenomenon at certain locations. This allows a control over the phenomenon which is essential in design and animation.

In this thesis we will focus mainly on two-scale models: the *global shape* and the *small-scale detail*. The global shape captures only the main features of the phenomenon. It emerges if we "blur" (remove the high frequencies) of the phenomenon. The global shape on its own is very smooth and makes the phenomenon look artificial, or man made. However,

as will be shown later in this chapter, this smooth shape will be very useful in the rendering process, such as the self-shadowing effect for clouds. The small scale detail is added to make the phenomenon look more "rough" or "complex". The user has control over the phenomenon by specifying the value of the global shape at certain locations.

One can argue that it is unnecessary to restrict oneself to the theory of random fields. That is, to consider only valid correlation measures and spectral density functions. As we are only interested in the visual characteristics of our model, another approach would be to create "ad hoc" functions for all kinds of phenomena. These functions are generally constructed by trial and error. More precisely, in this approach one starts with some function and then modifies it according to the visual result produced. This is the approach taken, for example, by Perlin [30]. The advantage of our more rigorous approach is that we can estimate correlation measures and spectral density functions from pictures of real phenomena using well established techniques. Furthermore there already exists abundant literature about such functions for many phenomena. Another advantage of our model is its generality. The same mathematical framework can be applied to a large variety of phenomena. Finally, the rigorous models can be analyzed more precisely. For example, aliasing problems can be avoided in advance if we know the spectral characteristics of our random field exactly.

4.2 The Global Shape

4.2.1 Smooth Estimation

The user constrains the global shape by specifying n scalar values d_i at the corresponding locations $\mathbf{t}_i$. The global shape constrained by these data can then be calculated by interpolation or by other approximation techniques. Exact interpolation may not seem to be a crucial condition in computer graphics; spline approximation techniques could also be used. However it turns out, as will be demonstrated, that exact interpolation has a simple mathematical formulation. Hence the value of the global shape at another location $\mathbf{t}$ is given by *smooth interpolation* from the given values. In smooth interpolation we look for a smooth function $L(\mathbf{t})$ such that

$$L(\mathbf{t}_i) = d_i \tag{4.1}$$

for $i = 1, \ldots, n$. Furthermore we require that the function is "well behaved" away from the data locations. It is well known that Lagrange interpolation can take huge values away from the data locations, hence that scheme is not suitable. Another approach is to use a so called *thin plate* interpolation [7] [27] [37]. This interpolation behaves like a thin plate constrained at the data locations which means that it tends to remain flat and hence is well behaved away from the data locations. A more general approach is to view the interpolation problem as an *estimation* problem. In (linear) estimation theory we wish to compute the value of a random field at a certain location $\mathbf{t}$, knowing (i) the random field at a set of locations $\mathbf{t}_i$ and (ii) the (second-order) statistics of the random field. In our case the known values are the d_i, and the second-order statistics are provided by a correlation measure.

Let us assume that the global scale is modelled by a homogeneous random field $R(\mathbf{t})$ with known second-order statistics. Therefore we can interpret the values d_i as the values of this field at the locations $\mathbf{t}_i$:

$$d_1 = R(\mathbf{t}_1), \ldots, d_n = R(\mathbf{t}_n). \tag{4.2}$$

Because we only consider second-order statistics it is sufficient to work with estimators of the form:

$$L(\mathbf{t}) = \sum_{i=1}^{n} \lambda_i d_i, \tag{4.3}$$

where the coefficients λ_i have to be determined. We impose two reasonable constraints on such estimators. First an estimator must be *correct* (*unbiased*):

$$E[L(\mathbf{t})] = E[R(\mathbf{t})] = \mu. \tag{4.4}$$

This corresponds to the equation:

$$\sum_{i=1}^{n} \lambda_i = 1. \tag{4.5}$$

This condition at most determines one of the unknowns λ_i. Therefore we need another constraint. A sensible one is to insist that the estimator have the minimum possible variance, in other words

$$E[(L(\mathbf{t}) - R(\mathbf{t}))^2] \tag{4.6}$$

is minimum over all such estimators. An estimator satisfying the above constraints is called a *best linear unbiased estimator* or in short BLUE. Here, "best" means minimum variance. In the next section a technique called *Kriging* will be described, which solves the above estimation problem.

4.2.2 Kriging

Let us first assume that the covariance of the random field $R(\mathbf{t})$ is well defined. It is important to note that in the development which follows it is *not* necessary to assume that the covariance function is isotropic. The minimality condition given by Equation 4.6 can be written in terms of the covariance function

$$C(\mathbf{0}) - 2\sum_{i=1}^{n} \lambda_i C(\mathbf{t} - \mathbf{t}_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j C(\mathbf{t}_j - \mathbf{t}_i). \tag{4.7}$$

This is a classic (constrained) minimization problem which can be easily solved by introducing a *Lagrange multiplier* ν. This procedure provides a system of $n+1$ linear equations known as the *Kriging system* [18]:

$$\sum_{j=1}^{n} \lambda_j C(\mathbf{t}_j - \mathbf{t}_i) - \nu = C(\mathbf{t} - \mathbf{t}_i), \qquad i = 1, \ldots, n \tag{4.8}$$

$$\sum_{j=1}^{n} \lambda_j = 1.$$

More simply, in matrix form the system is

$$\mathbf{M}\boldsymbol{\lambda} = \mathbf{b}(\mathbf{t}), \tag{4.9}$$

where $\mathbf{M}$ is an $(n+1) \times (n+1)$ matrix depending only on the data locations $\mathbf{t}_i$ and the covariance $C(\mathbf{h})$. More precisely, $\mathbf{M}$ is equal to the following matrix:

$$\mathbf{M} = \begin{pmatrix} C(\mathbf{0}) & C(\mathbf{t}_1 - \mathbf{t}_2) & \cdots & C(\mathbf{t}_1 - \mathbf{t}_n) & 1 \\ C(\mathbf{t}_2 - \mathbf{t}_1) & C(\mathbf{0}) & \cdots & C(\mathbf{t}_2 - \mathbf{t}_n) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C(\mathbf{t}_n - \mathbf{t}_1) & C(\mathbf{t}_n - \mathbf{t}_2) & \cdots & C(\mathbf{0}) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}. \tag{4.10}$$

$\mathbf{b}(\mathbf{t})$ is an $(n+1)$-dimensional vector depending on $\mathbf{t}$, the data locations $\mathbf{t}_i$ and the covariance $C(\mathbf{h})$ and is equal to

$$\mathbf{b}(\mathbf{t}) = (C(\mathbf{t} - \mathbf{t}_1), \ldots, C(\mathbf{t} - \mathbf{t}_n), 1)^t. \tag{4.11}$$

The vector $\boldsymbol{\lambda}$ is the unknown, and is given by

$$\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n, -\nu)^t. \tag{4.12}$$

At first glance this may look highly inefficient because we must solve a linear system for each location $\mathbf{t}$. However, as we will demonstrate, we have to solve one system for a given set of data values. Let us start by rewriting the estimator in vector form:

$$L(\mathbf{t}) = \boldsymbol{\lambda}^t \mathbf{d} \tag{4.13}$$

where $\mathbf{d}$ is equal to the "data" vector:

$$\mathbf{d} = (d_1, \ldots, d_n, 0)^t. \tag{4.14}$$

Because $\mathbf{M}$ is symmetric (see Equation 2.16) the estimator can be rewritten as:

$$L(\mathbf{t}) = \mathbf{b}(\mathbf{t})^t (\mathbf{M}^{-1})^t \mathbf{d} = \mathbf{b}(\mathbf{t})^t \mathbf{M}^{-1} \mathbf{d}. \tag{4.15}$$

Let $\mathbf{y}$ be the solution of the linear system:

$$\mathbf{M}\mathbf{y} = \mathbf{d}. \tag{4.16}$$

With this new vector the estimator can be rewritten as:

$$L(\mathbf{t}) = \mathbf{b}(\mathbf{t})^t \mathbf{y}. \tag{4.17}$$

As the system given in Equation 4.16 does not depend on the location $\mathbf{t}$, this means in terms of computer graphics that it only has to be solved at most once per frame. The dependence on the location is now included in the expression for the estimator (Equation 4.17):

$$L(\mathbf{t}) = \sum_{i=1}^{n} y_i C(\mathbf{t} - \mathbf{t}_i). \tag{4.18}$$

The linear system in Equation 4.16 has a unique solution if and only if the Kriging matrix is *strictly* positive definite and the data locations are mutually distinct. The first condition is satisfied if the covariance is strictly positive definite. Because the Kriging matrix is symmetric and positive definite, many stable numerical algorithms exist to solve the linear system such as the *Cholesky algorithm*. Instability may however arise if the data locations are very close together. One way to eliminate these instabilities is to replace a group of

nearby data points by a single point that has a value which is then a weighted average of the values at the previous data.

In case only the variogram is defined it is possible to obtain a similar linear system in terms of the variogram [18]. In practice, however, one works with the pseudo-covariance given by Equation 2.30, and solves the Kriging system for the pseudo-covariance function. Hence the global shape can be specified using only the variogram, and in particular it can be fractal.

If we consider a *Gaussian* isotropic correlation function, more precisely if

$$\rho(\mathbf{h}) = \exp(-\alpha\|\mathbf{h}\|^2), \tag{4.19}$$

then the estimator of Equation 4.18 is an instance of Blinn's blobbies [5]. Hence, for Gaussian correlation functions, the model of the global shape can be viewed as a generalization of a blobby.

4.2.3 Extensions

The above Kriging scheme only applies to random fields that have a constant mean (see Equation 4.4). Many natural phenomena, however, have a mean which varies over space or time. In the geostatistical literature this non-constant mean is referred to as a *drift*. In case the drift is known, it is possible to introduce the new random variable:

$$Y(\mathbf{t}) = R(\mathbf{t}) - \mu(\mathbf{t}). \tag{4.20}$$

This new variable Y has the same covariance as the original random variable R and has zero mean. Hence we can apply the above Kriging scheme to the variable Y, which produces an estimator $K(\mathbf{t})$. The final estimator for R is then given by:

$$L(\mathbf{t}) = \mu(\mathbf{t}) + K(\mathbf{t}). \tag{4.21}$$

This works well if the drift is known exactly, but unfortunately this is rarely the case in practice. When the drift is unknown, one approach is to assume it has a "simple" form, such as a linear combination of some simple basis functions $f_l(\mathbf{t})$:

$$\mu(\mathbf{t}) = \sum_{l=1}^{k} a_l f_l(\mathbf{t}), \tag{4.22}$$

where the a_l are unknown coefficients which have to be estimated as well. In three dimensions, it is usual to consider *quadratic drifts* [18]:

$$\mu(x,y,z) = a_1 + a_2x + a_3y + a_4z + a_5x^2 + a_6xy + a_7xz + a_8y^2 + a_9yz + a_{10}z^2. \tag{4.23}$$

The unknowns a_i now become part of the Kriging system, and the unbias condition is:

$$E[R(\mathbf{t}) - L(\mathbf{t})] = \sum_{l=1}^{k} a_l \left(f_l(\mathbf{t}) - \sum_{i=1}^{n} \lambda_i f_l(\mathbf{t}_i) \right) = 0. \tag{4.24}$$

The following equations are a sufficient condition for the above expression to vanish:

$$\sum_{i=1}^{n} \lambda_i f_l(\mathbf{t}_i) = f_l(\mathbf{t}), \qquad l = 1, \ldots, k. \tag{4.25}$$

This constraint is k-dimensional, as opposed to the 1-dimensional constraint of Equation 4.5. Hence to solve the variational problem we now have to introduce k Lagrange multipliers $\nu_1, \ldots, \nu_k$. This leads to the following equations (cf. Equation 4.8):

$$\sum_{j=1}^{n} \lambda_j C(\mathbf{t}_j - \mathbf{t}_i) - \sum_{l=1}^{k} \nu_l f_l(\mathbf{t}_i) = C(\mathbf{t} - \mathbf{t}_i), \qquad i = 1, \ldots, n. \tag{4.26}$$

As before, these equations can be written as a linear system:

$$\mathbf{M}\boldsymbol{\lambda} = \mathbf{b}(\mathbf{t}). \tag{4.27}$$

The Kriging matrix is now given by:

$$\mathbf{M} = \begin{pmatrix} C(\mathbf{0}) & \cdots & C(\mathbf{t}_1 - \mathbf{t}_n) & 1 & f_2(\mathbf{t}_1) & \cdots & f_k(\mathbf{t}_1) \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ C(\mathbf{t}_n - \mathbf{t}_1) & \cdots & C(\mathbf{0}) & 1 & f_2(\mathbf{t}_n) & \cdots & f_k(\mathbf{t}_n) \\ 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ f_2(\mathbf{t}_1) & \cdots & f_2(\mathbf{t}_n) & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_k(\mathbf{t}_1) & \cdots & f_k(\mathbf{t}_n) & 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{4.28}$$

and the vector $\mathbf{b}(\mathbf{t})$ is equal to

$$(C(\mathbf{t} - \mathbf{t}_1), \ldots, C(\mathbf{t} - \mathbf{t}_n), 1, f_2(\mathbf{t}), \ldots, f_k(\mathbf{t}))^t, \tag{4.29}$$

while the vector of unknowns is now:

$$\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n, -\nu_1, -\nu_2, \ldots, -\nu_k)^t. \tag{4.30}$$

As in the constant mean case, the matrix $\mathbf{M}$ is symmetric. In order for the system to have a unique solution, we must add the constraint that the basis functions $f_l(\mathbf{t})$ are linearly independent on the set of the n data points $\mathbf{t}_i$ [18]. Using the same argument as before we can prove that such a system has to be solved at most once per frame.

Other extensions of the Kriging scheme are mentioned in [18]. If the set of data points gets too large, it is possible to solve several Kriging systems on subsets of the data, and then combine the results to obtain a single estimator. This can however make the estimator discontinuous [18]. When the random field itself is vector valued (as in modelling wind for example) then it is possible to estimate each component at the same time, taking into account the cross-correlations between the components. The latter estimation procedure is referred to as *coKriging*. The uncertainty at the data locations can also be incorporated into the Kriging scheme [11].

4.3 Small Scale Detail

4.3.1 Random Functions

We shall model the small scale detail by a random field $R(\mathbf{t})$ with given second-order statistics. In the spirit of Lewis' work [24] we want a *model-directed synthesis* function $W(\mathbf{t})$ that approximates the desired random field. More precisely, we want to be able

to evaluate the function at arbitrary locations in space, as opposed to random functions generated by the spectral methods discussed in the previous chapter, that have values which are defined only on a regular grid. The location of evaluation should be determined by the model and not by the noise synthesis technique. For example, we may want to evaluate the random function only near the boundary of the global smooth model. Moreover we want the evaluation of the function to be independent of previous calculations which is not the case, for example, in stochastic subdivision. This allows us to evaluate the random field at different locations simultaneously, and hence permits a straightforward synthesis of the model in parallel. Typically the function $W(\mathbf{t})$ is constructed by summing more elementary random functions at different scales (frequencies).

4.3.2 Spectral Sums

As stated in Chapter 2, random fields can be arbitrarily approximated closely by sums of simple random functions (cf. Equation 2.41). This is the basic idea behind various spectral sum random functions. Hence the spectral density function is used to specify the second-order statistics of the random field. The random function $W(\mathbf{t})$ is then given by:

$$W(\mathbf{t}) = \sum_{i=N_{min}}^{N_{max}} A_i W_i(\mathbf{t}). \tag{4.31}$$

In other terms, W is a summation of band-limited random functions W_i. Each of these functions has a very narrow spectrum and ideally we would like all of these spectra to be mutually disjointed. The A_i together model the spectral density function, i.e.,

$$S(\omega) \approx A_i \qquad a_i < \omega < b_i \tag{4.32}$$

for all frequencies ω in the spectrum $[a_i, b_i]$ of W_i, that is in a neighbourhood of ω_i. The bounds of the summation N_{min} and N_{max} determine the "quality" of the random function. The more terms we include the better the function will approximate the random field, but the computation costs will increase as well. Thus in practice the values of these bounds are a tradeoff between image quality and efficiency.

4.3.3 Perlin's Noise

One of the most popular classes of random functions used in computer graphics is that introduced in [30] by Perlin. An approximation of white noise $N(\mathbf{t})$ is used as the basic random function from which more complex functions are built. The function $N(\mathbf{t})$ is an interpolator of an ensemble of uncorrelated random values given on an integer lattice. Several possible interpolation schemes are discussed in [24]. The interpolator has the effect of a low-pass filter, hence the function $N(\mathbf{t})$ is nearly band limited. Perlin's function is a sum of scaled versions of this basic noise function:

$$W(\mathbf{t}) = \sum_{i=N_{min}}^{N_{max}} \frac{1}{2^i} N(2^i \mathbf{t}). \tag{4.33}$$

Perlin claims that this is an approximation of a fractal with spectral exponent $\beta = 1$. Saupe in [35] on the other hand empirically determines the exponent to be $\beta = 3$. The problem with this function lies in the spectra of the scaled versions of N. The $N(2^i \mathbf{t})$ are

not mutually disjointed since they all include the lower frequencies because of the "low pass' effect of the interpolation [24]. If the resolution of the image is r, then safe choices for the bounds of the sum are $N_{min} = \log(1/r) - 1$ and $N_{max} = \log(r) + 1$. Therefore the number of terms in the sum should be approximately $2\log(r)$.

4.3.4 Gardner's Texture

In order to render his clouds, Gardner introduces the following texture functions [13]:

$$W(x, y, z) = k \sum_{i=1}^{n} F_i(x, y, z) \sum_{i=1}^{n} F_i(y, x, z) \tag{4.34}$$

where the F_i are defined by

$$F_i(u, v, z) = C_i \sin(\omega_i u + (\pi/2) \sin(\omega_{i-1} v) + \pi \sin(\omega_i z/2)). \tag{4.35}$$

The C_i are coefficients, and the ω_i are characteristic frequencies. Gardner achieves "realistic looking" pictures by choosing the following values for the C_i and ω_i:

$$C_i = \left(\frac{1}{\sqrt{2}}\right)^i C_0 \tag{4.36}$$

$$\omega_i = 2^i \omega_0. \tag{4.37}$$

Gardner claims that these values produce a $1/f$ fractional noise. However, a more careful analysis shows that the spectral density function is more complicated. The series expansion of the functions F_i is actually (discarding the z argument) [4]:

$$F_i(u, v) = \sum_{n=0}^{\infty} J_n(\pi/2)[\sin(2^{i-1}\omega_0(2u + nv)) + (-1)^n \sin(2^{i-1}\omega_0(2u - nv))]. \tag{4.38}$$

Hence Gardner's function is an approximation of $1/f$ fractional noise only when $u = kv$, for any integer k. Note that this function has no random component and the apparent randomness comes entirely from the coupling of the phases in the functions F_i.

4.3.5 The Weierstrass-Mandelbrot Function

A more rigorous approximation of a fractal is given by the *Weierstrass-Mandelbrot* function. This is a generalization of the "nowhere differentiable but everywhere continuous" function of Weierstrass. An fBm with fractal parameter H is approximated by a superposition of sinusoids with geometrically spaced frequencies [25]:

$$W(t) = \sum_{n=-\infty}^{+\infty} r^{Hn} \cos(r^{-n} t + \phi_n) \tag{4.39}$$

where the ϕ_n are independent and uniformly distributed over $[0, 2\pi]$ and r is a parameter referred to as the *lacunarity parameter* [41]. Berry and Lewis in [3] show that the statistics of this random function closely approximate an fBm. Because the sum includes terms for arbitrarily large and small frequencies, the function has no characteristic scale. As for Perlin's noise function, only approximately $2\log(r)$ terms have to be considered for visually "good" results. This is why the Weierstrass-Mandelbrot function is preferred to a (discrete) Fourier summation which contains the order of r terms. The storage costs are thus clearly lower in general. The use of a periodic function other than $\cos(x)$, such as the triangle function, give functions which are visually similar using the same number of terms.

4.3.6 Superposition of One-Dimensional Functions

Ausloos and Berman in [2] generalize the Weierstrass-Mandelbrot function to higher dimensions. The most straightforward generalization is to replace the one-dimensional argument t by the norm of the multi-dimensional argument $\mathbf{t}$:

$$W(\mathbf{t}) = W(\|\mathbf{t}\|). \tag{4.40}$$

However, this does not provide a function that approximates an fBm [2], and because the function is constant on circles (spheres) of a given radius, this produces images which are too regular. Another approach is to consider a summation of terms of the form $\cos(x)\cos(y)$. This results in strong directional artifacts and is unsuited for random functions [24]. Instead, Ausloos and Berman propose the following generalization to two dimensions:

$$W(x,y) = \sum_{m=1}^{M} A_m W_m(a_m x + b_m y) \tag{4.41}$$

where the W_m are one-dimensional Weierstrass-Mandelbrot functions and (a_m, b_m) specify a direction. The resulting function $W_m(a_m x + b_m y)$ is thus a ridge-like surface, i.e. a one-dimensional signal dragged along a given direction. Furthermore they show that this function is equivalent to Mandelbrot's construction of multi-dimensional fBms [25]. This approach can be generalized to other random functions. Let $X_1(t), \ldots, X_M(t)$ be an ensemble of M one-dimensional random functions with given second-order statistics. Therefore a two-dimensional random field can be defined by:

$$W(x,y) = \sum_{m=1}^{M} A_m X_m(a_m x + b_m y). \tag{4.42}$$

The one-dimensional random functions can be generated, for example, by spectral methods using the spectral density function as a modelling tool. Anjyo's random functions in [1] are very close to this formulation. The second-order statistics of $W(x,y)$ can be determined from the statistics of the one-dimensional functions. However, it is not clear how to choose the statistics of the one-dimensional functions to get the desired second-order statistics of $W(x,y)$. Hence the function $W(x,y)$ is not directly modelled by its second-order statistics, but instead by its functional relation to the functions X_i.

Nothing stops us from generalizing this approach to three-dimensional random fields. For example, the 3-D function can be modelled as a superposition of one-dimensional signals in a particular direction, or as a superposition of two-dimensional signals, or a blend of both.

4.3.7 Sparse Convolution

In Section 5 of Chapter 2 we reviewed how to generate a random field by filtering a "simple" random field such as white noise. Lewis in [24] considers another canonical random field: the *Poisson noise process*. This random field $P(\mathbf{t})$ consists of a sum of impulses of uncorrelated intensity distributed at uncorrelated locations in space:

$$P(\mathbf{t}) = \sum_{i=1}^{N} a_i \delta(\mathbf{t} - \mathbf{t}_i). \tag{4.43}$$

Convolving this field with a kernel $h(\mathbf{t})$ synthesizes the following random function:

$$W(\mathbf{t}) = \sum_{i=1}^{N} a_i h(\mathbf{t} - \mathbf{t}_i). \tag{4.44}$$

As the spectral density of the Poisson process $P(\mathbf{t})$ is clearly constant, the spectral density of the resulting function $W(\mathbf{t})$ is equal to the square root of the spectral density of the convolution kernel (see Equation 2.57). The advantage of this approach is that the quality of the noise can be controlled by varying the number of impulses $\mathbf{t}_i$. Lewis furthermore discusses several techniques to speed up the computation of this function by using lookup tables for the kernel $h(\mathbf{t})$.

4.4 Stochastic Kriging

The above two models can be combined in the Kriging procedure. This technique was first suggested by Journel in [17] under the name of *conditional simulation*. The procedure has been applied to other fields thereafter by Delhomme in the Hydrosciences [11] and by Hewett et al. in Petroleum Engineering [15] [16].

The idea behind stochastic Kriging is very simple. If $R(\mathbf{t})$ is the random field we want to simulate, and $\hat{R}(\mathbf{t})$ is the estimator obtained from the Kriging system, then we can write:

$$R(\mathbf{t}) = \hat{R}(\mathbf{t}) + \left(R(\mathbf{t}) - \hat{R}(\mathbf{t}) \right). \tag{4.45}$$

In other words the unknown random field is the sum of the Kriging estimate and the Kriging error. Unfortunately the Kriging error remains unknown because it involves the function $R(\mathbf{t})$. The solution proposed by Journel is to consider any known realization $S(\mathbf{t})$ of a random field independent of $R(\mathbf{t})$ with the same covariance as the unknown random field $R(\mathbf{t})$. Since the random field $S(\mathbf{t})$ is known, we can calculate the Kriging estimate $\hat{S}(\mathbf{t})$ which interpolates $S(\mathbf{t})$ at the user-supplied data locations. We now consider the function:

$$R^*(\mathbf{t}) = \hat{R}(\mathbf{t}) + \left(S(\mathbf{t}) - \hat{S}(\mathbf{t}) \right). \tag{4.46}$$

In the above expression all terms are known. In practice, the function $S(\mathbf{t})$ can be calculated by using any of the techniques mentioned in the previous section. It can be shown that the second-order statistics of the function $R^*(\mathbf{t})$ are indistinguishable from those of the unknown random field $R(\mathbf{t})$ [17]. In reference to the global scale, only one Kriging system has to be solved. However, we have to add the value of the function $S(\mathbf{t})$ evaluated at the data locations to the data constraints. More explicitly, the previous data constraints $(\mathbf{t}_i, d_i)$ become $(\mathbf{t}_i, d_i - S(\mathbf{t}_i))$. Let us use $K(\mathbf{t})$ to denote the estimator obtained from this system. Then our final function is given by:

$$R(\mathbf{t}) = K(\mathbf{t}) + S(\mathbf{t}). \tag{4.47}$$

Hence, this method is not more complicated than the two-scale model previously presented. However the method has certain drawbacks. Most importantly the second-order statistics of the global shape and the small-scale detail have to be identical. This works when one is dealing with fractals which have no characteristic scale [15] [16]. In computer graphics, however, we can sacrifice rigour for modelling power by allowing the small scale detail function $S(\mathbf{t})$ to have any second-order statistics. Furthermore we can add a parameter λ which controls the amount of small scale noise added:

$$R(\mathbf{t}) = K(\mathbf{t}) + \lambda S(\mathbf{t}). \tag{4.48}$$

4.5 Possible Extensions

In this section we will discuss possible extensions of the model to more than two scales. Let us assume that we want L different scales. Then, the phenomenon will be modelled by a random field given by a sum of random fields R_l at the different scales:

$$R(\mathbf{t}) = \sum_{l=1}^{L} R_l(\mathbf{t}). \tag{4.49}$$

Each random field is specified by its second-order statistics and (for some levels) by user-supplied data. The user thus has control over a wide range of scales. It is however unclear how the user would specify this data. One possibility is a *top-down* design process. At first the user specifies the shape of the the largest scale by providing a set of data values. He or she then views a display of the interpolated shape (obtained by the Kriging procedure), and starts to work on the next scale level by specifying refinements of the previously-computed shape. From these refinements, the function R_2 is calculated. Of course the statistics of R_2 should not interfere with those of R_1. After this step the user views the sum of both fields and starts refining it for the next scale level to produce R_3 and so on. At each level a certain amount of noise can be added to add more complexity to the model. This is generated in the same way as the small-scale detail of the two-scale model is generated by providing a random function W_l at each level, with specified second-order statistics.

At present it is still unclear how this design process can be formalized. Similar multi-scale models have already been developed in other fields, however. In low-level vision, Szeliski [38] presents several models to separate an image into different scales of description. The techniques he uses cost too much in memory requirements for three-dimensional phenomena. In image compression, Burt and Adelson [9] present a way to compress images more efficiently by building a multi-scale representation of the image. This approach has many attractive features which make it a candidate for a multiple-scale model in computer graphics. A similar approach has been proposed in computer graphics by Williams [42] to encode a texture at different scales.

A formalization of the procedure might be in terms of a hierarchy of Kriging systems at each level. The technical details remain to be worked out, however. Furthermore, as the distances between the data points decrease at each level, one has to be careful about numerical instabilities which may arise. Another difficulty is the increase in the number of data points at each level. This could be resolved by solving several Kriging systems and then combining the resulting estimators.

Chapter 5

Application of the Model to Clouds

> I truly do want to know how to describe clouds. But to say there's a piece over here with that much density, and next to it a piece with this much density – to accumulate that much detailed information, I think is wrong. It's certainly not how a human being perceives those things, and it's not how an artist perceives them.
> *Mitchell Feigenbaum*

In this chapter we will provide a specific example of application of our model to natural phenomena: the modelling of clouds. The previous chapters dealt primarily with the modelling of phenomena; in this chapter we will consider the rendering process as well.

Clouds are a typical example of partially translucent phenomena. As well, the visual simulation of clouds is important in many applications. One such application is flight-simulation. We will model clouds as an ellipsoidal random field with two characteristic scales. In the next section we will review some visual properties and terminology of clouds. In the subsequent sections we will discuss various rendering techniques. We will begin with a brief review of the theory of radiative transfer, which is the physical theory of how light is distributed within a cloud. Several simple models will then be reviewed. A section will be devoted to the ray-tracing of the global shape, which is similar to ray-tracing blobbies.

5.1 Observed Properties of Clouds

A vast amount of literature deals with the visual properties of clouds. This research was initiated by the work of the British pharmacist Luke Howard in 1803. Howard classified clouds into four major types: *cumulus* (cauliflower type clouds, which have vertical development), *stratus* (clouds in flat appearing layers), *cirrus* (fibrous or hair-like), and *nimbus* (rain clouds). Of course, some real clouds are hybrids, such as the *cumulo-nimbus* clouds which can be observed during thunder-storms.

Basically, cloud phenomena have three distinct scales [32]: the *masoscale* describes clouds of the range of 1000 km and up (the cloud patterns viewed from a satellite for example), the *mesoscale* includes clouds ranging from a few kilometres to several hundred kilometres. In this range are the clouds listed in Howard's taxonomy. The smallest scale is the *microscale* which describes cloud phenomena on a scale smaller than 1 km (for example the cloud "puffs", the turrets on cumulus clouds and other small scale irregularities).

Recently Cahalan [10] has proposed a multi-scale fractal model to simulate clouds. The model is used to recover the density distribution of clouds from experimental data. He

considers a two-scale model. The large scale models the distribution of isolated clouds within cloud clusters. The small scale describes the characteristics of an isolated cloud. Each scale is modelled as an fBm with a spectral exponent β that is estimated from the experimental data obtained by satellites.

5.2 The Scattering Equation

The rendering of a cloud as an ensemble of particles can be solved (in theory) by using the equations of *radiative transfer* [22]. The quantity calculated in a scattering problem is the *radiance* $I(\mathbf{x}, \mathbf{s})$ at a given position $\mathbf{x}$ emitted in the direction given by the solid angle $\mathbf{s}$. The phase function $p(\mathbf{x}, \mathbf{s}, \mathbf{s}')$ characterizes the scattering in direction $\mathbf{s}$ of radiance arriving from the direction $\mathbf{s}'$ at location $\mathbf{x}$. The phase function is normalized by:

$$\int_{\|\mathbf{s}'\|=1} p(\mathbf{x}, \mathbf{s}, \mathbf{s}') d\mathbf{s}' = 4\pi. \tag{5.1}$$

The total amount of radiance emanating from a particular direction $\mathbf{s}$ at a location $\mathbf{x}$ is given by the *source function*:

$$J(\mathbf{x}, \mathbf{s}) = \frac{\alpha(\mathbf{x})}{4\pi} \int_{\|\mathbf{s}'\|=1} p(\mathbf{x}, \mathbf{s}, \mathbf{s}') I(\mathbf{x}, \mathbf{s}')\, d\mathbf{s}' + J_s(\mathbf{x}, \mathbf{s}). \tag{5.2}$$

The function $\alpha(\mathbf{x})$ is the *albedo* of the cloud at location $\mathbf{x}$. The function $J_s(\mathbf{x}, \mathbf{s})$ is a term which accounts for internal and/or external sources of radiation. The radiance I can be calculated from the *scattering equation* [22]:

$$\mathbf{s} \cdot \nabla_{\mathbf{x}} I(\mathbf{x}, \mathbf{s}) = -\sigma_e(\mathbf{x})[I(\mathbf{x}, \mathbf{s}) - J(\mathbf{x}, \mathbf{s})] \tag{5.3}$$

with some boundary conditions. The function σ_e is the *extinction* coefficient and is equal to

$$\sigma_e(\mathbf{x}) = \sigma_a(\mathbf{x}) + \sigma_s(\mathbf{x}) \tag{5.4}$$

where σ_a is the *absorption* coefficient and σ_s is the *scattering* coefficient. The scattering equation states that the variation of the radiation in direction $\mathbf{s}$ is simply the difference between the emitted radiance and the absorbed radiance. This equation can only be solved in practice in particularly simple cases where the dimensionality of the radiance I is lowered. Furthermore, the radiance function (if calculated) provides too much information for computer graphics. We are not interested in the intensity values of the centre of a cloud, since we cannot see such points. In many cases it is assumed that the cloud is isotropic. More precisely, if the phase function only depends on the angle θ between the two directions $\mathbf{s}$ and $\mathbf{s}'$:

$$p(\mathbf{x}, \mathbf{s}, \mathbf{s}') = p(\mathbf{x}, \cos\theta). \tag{5.5}$$

5.3 Low Albedo Approximation

Blinn [6] and Kajiya and Von Herzen [20] present algorithms to ray-trace volume densities with a low albedo. This means that multiple scattering within the cloud, i.e. particles illuminating other particles of the cloud, are ignored. First the cloud is modelled as a density map $\rho(\mathbf{x})$. The phase function is then rewritten as:

$$p(\mathbf{x}, \cos\theta) = \rho(\mathbf{x}) p(\cos\theta). \tag{5.6}$$

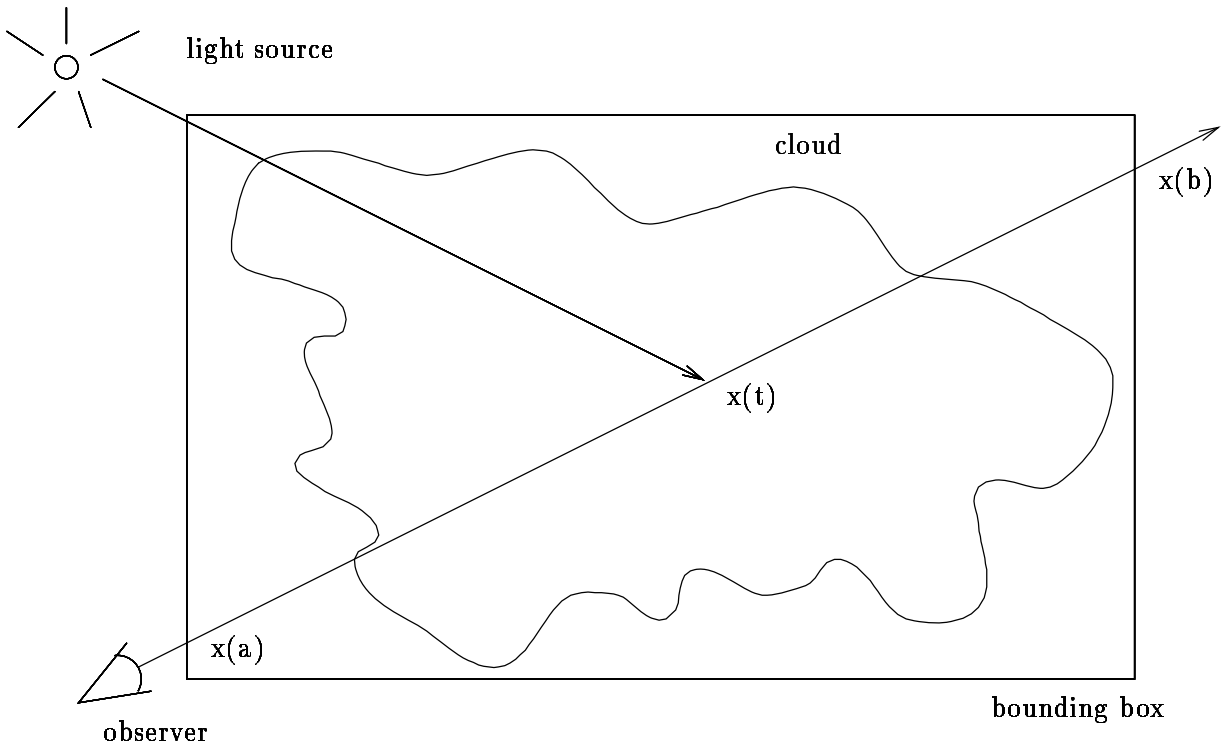

Figure 5.1: Low Albedo Approximation

Assuming we consider only the radiance coming from the line $(x(t), y(t), z(t))$, then the scattering equation with the low albedo approximation becomes:

$$\frac{d}{dt} I(x(t), y(t), z(t)) = -\sigma_e \rho(x(t), y(t), z(t)) I(x(t), y(t), z(t)). \tag{5.7}$$

This has solution of

$$I(t) = I(t_0) \exp\left(-\sigma_e \int_{t_0}^{t} \rho(x(\tau), y(\tau), z(\tau))\, d\tau\right), \tag{5.8}$$

i.e., a simple exponential decay. The algorithm to ray-trace clouds is based on this exponential decay [20]. Let us assume that we have L light sources. We first consider a point $\mathbf{x}(t)$ on the ray within the cloud. The amount of light reaching $\mathbf{x}(t)$ from a light source i is then given by:

$$\Lambda_i(t) = \Lambda_i \exp\left(-\sigma_e \int \rho(\tau)\, d\tau\right). \tag{5.9}$$

The integral is over a line segment from $\mathbf{x}(t)$ to the location of light source i and Λ_i is the intensity of the light source. The particles on a portion dt of the ray contribute the following amount to the final illumination value:

$$dI_i = \Lambda_i(t) \sigma_s \rho(\mathbf{x}(t)) p(\cos\theta_i(t)) \exp\left(-\sigma_e \int_a^t \rho(\mathbf{x}(\tau))\, d\tau\right)\, dt, \tag{5.10}$$

where $\theta_i(t)$ is the angle between the ray and the segment from $\mathbf{x}(t)$ to the light source i. The final intensity I is then given by summing over all light sources and contributions dI_i:

$$I = \sum_{i=1}^{L} \int_a^b dI_i \; + \; I_{back} \exp\left(-\sigma_e \int_a^b \rho(\mathbf{x}(t))\, dt\right). \tag{5.11}$$

The intensity I_{back} is the light coming from behind the cloud. The bounds of the integral a and b are given by the intersection of the ray with a volume bounding the cloud. See Figure 5.1 for a better understanding of the terms in the above equation. As noted in [20] the values $\Lambda_i(t)$ can be interpolated from a precomputed table of values at discrete locations of space. Blinn was able to solve the integrals involved analytically because of the simple geometry of his clouds. In the general case, however, these integrals can become very expensive. Sakas [34] ray-traces density maps given as a voxel data base. The integrals are then evaluated using voxel traversal algorithms. This method can also account for self-shadowing effects which are especially important in the case of thick cumulus clouds by shooting additional rays. However, as noted [20] this can result in pictures which are overly obscured because the self-scattering effect is neglected (due to the low albedo assumption).

5.4 The Rendering Algorithm

We will now introduce the rendering algorithm which takes advantage of the double-scale model presented in this thesis. The rendering algorithm dealt with in the previous section suffers mainly from an inability to account for self-shadowing and self-scattering effects, which are semi-global. In the following algorithm these effects are accounted for by using the geometry of the global shape of the model. Furthermore, by having analytic expressions for the density map we can write special code to solve the integrals. An analytic solution is not possible because of the addition of noise. It is also possible to model the translucence of the cloud by using a modification of Gardner's algorithm [13]. The general outline of the algorithm is as follows:

For each ray do
(1) Calculate intersection point with isosurface of global shape
(2) If no intersection then next ray
(3) Calculate normal at intersection point
(4) Use normal to calculate self-shadowing
(5) Determine brightness and translucence of the cloud
Next ray

In the next sections we will describe the intersection step (1) more carefully and give a simple algorithm based on Gardner's work for step (5).

5.5 Ray Tracing Blobbies

The intersection step of a ray and a blobby will now be described in more detail. Rather than use Blinn's algorithm [5], we will use instead *interval arithmetic* [28]. Since we have assumed that the random field defining the global shape is ellipsoidal, the estimator given by the Kriging system will have the following form:

$$L(\mathbf{t}) = \sum_{i=1}^{n} y_i C((\mathbf{t} - \mathbf{t}_i)^t \mathbf{Q}_i (\mathbf{t} - \mathbf{t}_i)). \tag{5.12}$$

The matrices $\mathbf{Q}_i$ will be as defined in section 2.2.2. The blobby will then be defined as an isosurface of this function:

$$L(\mathbf{t}) - T = 0 \tag{5.13}$$

where T is a given threshold.

When calculating the intersection of a ray $R(t)$ and the blobby, first we transform the ray to its canonical form

$$R(t) = (0, 0, t) \tag{5.14}$$

To find the intersection we substitute $R(t)$ into $L(\mathbf{t})$ to get:

$$\varphi(t) = \sum_{i=1}^{n} y_i C(c_i + b_i(t - z_i) + a_i(t - z_i)^2) \tag{5.15}$$

where a_i, b_i and c_i are given by the coefficients of $\mathbf{Q}_i$ and the components of the locations $\mathbf{t}_i$. The intersection points of the ray with the blobby are given by the roots of the equation:

$$\varphi(t) - T = 0. \tag{5.16}$$

To find these intersections we first compute a set of intervals. Each interval contains exactly one root. This is done by interval arithmetic. Then a standard root finder is applied to each interval which is guaranteed to converge because the interval only contains a single root. An example of such root finding algorithm is the well-known *Newton iteration*. Now we will describe how interval arithmetic is applied to this particular problem.

Interval arithmetic is an extension of standard arithmetic to intervals. In other words we define operations (such as addition and multiplication) on intervals. We make the following definitions:

$$[a, b] + [c, d] = [a + c, b + d] \tag{5.17}$$

$$c[a, b] = [ca, cb] \;\; if\; c > 0 \tag{5.18}$$

$$c[a, b] = [cb, ca] \;\; if\; c < 0 \tag{5.19}$$

$$\exp([a, b]) = [\exp(a), \exp(b)] \tag{5.20}$$

More generally, for any non-decreasing function $f(x)$ and non-increasing function $g(x)$ we must

$$f([a, b]) = [f(a), f(b)] \tag{5.21}$$

$$g([a, b]) = [g(b), g(a)] \tag{5.22}$$

In the case of non-monotonic functions the definition is less straightforward and we have to know all the local extrema of the function in advance in order to identify the intervals on which the function is monotonic. A specific example will be given later.

We are searching for intervals $[t_1, t_2]$ such that $T \in \varphi([t_1, t_2])$ (the interval contains a root) and the derivative $\varphi'([t_1, t_2])$ does not contain 0 (the function is monotonic in the interval). These two facts imply that the interval contains a single root. The following algorithm starts with an initial interval $[t_1, t_2]$ (provided for example by some bounding box heuristics), and then proceeds by recursively subdividing the interval in two. The intervals are contained in the set I at the end of the algorithm. More explicitly:

```
I := ∅
miss := TRUE
Algorithm Isolate([t1, t2])
        if T ∉ φ([t1, t2]) then
          return
        if |t2 − t1| < ε then
          I := I ∪ {[t1, t2]}
          miss := FALSE
          return
        if 0 ∉ φ'([t1, t2]) then
          if (φ(t1) − T)(φ(t2) − T) < 0 then
            I := I ∪ {[t1, t2]}
            miss := FALSE
          return
        t0 := (t1 + t2)/2
        Isolate([t1, t0])
        Isolate([t0, t2])
        return
end
```

We now consider an explicit example where the covariance function is Gaussian, i.e. of the form:

$$C(t) = \exp(-\alpha t^2). \tag{5.23}$$

From Equations 5.17, 5.18 and 5.19, we know how to calculate weighted sums of intervals. More precisely, if we assume that the coefficients y_i are positive, then

$$\varphi([t_1, t_2]) = \sum_{i=1}^{n} y_i f_i([t_1, t_2]). \tag{5.24}$$

Therefore we need only show how to compute the functions f_i, which are given explicitly by:

$$f(t) = f_i(t) = \exp(-\alpha(c + b(t - z_i) + a(t - z_i)^2)) \tag{5.25}$$

See Figure 5.2 for a plot of this function. Let the interval to be evaluated be $[t_1, t_2]$. If $f(t_1) > f(t_2)$ then we swap the values of t_1 and t_2. The function has a unique maximum at $t_{max} = z_i - b/(2a)$. Therefore if the interval $[t_1, t_2]$ contains t_{max}, then we set the resulting interval to:

$$[f(t_1), f(t_{max})] = [f(t_1), f(c - b^2/(4a))]. \tag{5.26}$$

If not, the function $f(t)$ is monotonic between t_1 and t_2, and we simply return:

$$[f(t_1), f(t_2)]. \tag{5.27}$$

The evaluation of the derivative $\varphi'(t)$ is a bit more complicated. As above it is sufficient to show the calculation only for a single term of the sum:

$$f'(t) = -\alpha(b + 2a(t - z_i))f(t) \tag{5.28}$$

See Figure 5.3 for a plot of this function. This function has a minimum and a maximum at the following points:

$$t_{\pm} = z_i - \frac{b}{2a} \pm \frac{1}{\sqrt{2a\alpha}}. \tag{5.29}$$

Figure 5.2: The Correlation Function

Figure 5.3: Derivative of the Correlation Function

Then as before we check to see if the input interval contains any of the extrema, and we return specific values accordingly:

$A = f'(t_1)$
$B = f'(t_2)$
if $A > B$ then
  swap the values of A and B
end if
if $t_- \in [t_1, t_2]$ then
  $B = f'(t_-)$
end if
if $t_+ \in [t_1, t_2]$ then
  $A = f'(t_+)$
end if

The resulting interval is then given by $[A, B]$. A safety trick to avoid problems caused by numerical imprecision is to make the resulting intervals slightly larger than given by the above algorithms, for example by an amount ϵ.

5.6 A Simple Illumination Model

In this section we will present an illumination model which is similar to Gardner's textured ellipsoids algorithm [13]. We assume that we already have the nearest intersection point P of the ray and the blobby, and the normal N at that point. From these values a "classic" illumination value $I_{classic}$ can be calculated (for example using Phong's model). This value will be used to account for the the brightness and the self-shadowing effects of the cloud. The translucence is obtained from the small scale detail function $W(\mathbf{t})$, which is used as a (solid) texture. The texture value is given by $t = W(P)$. This value is compared to a threshold T which is a function of the distance d traversed by the ray through the cloud and the cosine μ of the angle between the viewing vector and the normal N. Both a threshold T_1 at the "boundary" of the cloud (typically very high) and a threshold T_2 at the "centre" of the cloud (typically very low) are defined. The threshold T is then interpolated from these two values as follows:

$$T = T_1 + (T_2 - T_1)\exp(-\beta d\mu^2) \tag{5.30}$$

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

$$t_0 = \max(0, \nu(t, T)) \tag{5.31}$$

where ν is some normalization function, which in our implementation was set to

$$\nu(t, T) = (t - T)/(1 - T). \tag{5.32}$$

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

$$I = t_0 I_{classic} + (1 - t_0)\exp(-\beta d\mu^2) I_{back}. \tag{5.33}$$

Note that this illumination model is not based on the physical equations of radiative transfer theory. It was designed simply to give visually good results.

Chapter 6

Results

In this chapter we will demonstrate the power of our two-scale model with some actual examples. The next section will describe the different options in the modelling process using our model. The second section will present the actual implementation of the simple illumination model for the three-dimensional cloud model.

6.1 The Modelling Process

For the sake of clarity we will only consider two-dimensional phenomena in this section. The images can, for example, represent thin (plane parallel) cloud layers. The goal in this section is not to portray realistic pictures, but rather to exhibit the flexibility of our model.

6.1.1 Specification of the Global Shape

In the first stage of the modelling process the user specifies a set of values at data points. This way she or he roughly sketches the shape of the phenomenon. Figure 6.1 shows a set of values which are magnified for more clarity. The exact shape of the globally smooth model now depends on the choice of the correlation function. We will now exhibit images for different correlation functions. First we consider the case of an isotropic Gaussian correlation function:

$$\rho(x, y) = \exp\left(-\alpha(x^2 + y^2)\right). \tag{6.1}$$

The parameter α determines the range of the correlation: high values mean almost no correlation between the data points and low values mean a high correlation between the points. Figures 6.1, 6.2 and 6.3 show images synthesized for different values of α along with a plot of the corresponding correlation function. The isotropy of the correlation function is clearly visible. To introduce anisotropies, the user can work with an ellipsoidal correlation function by specifying a "stretch" or a "squeeze" along two directions (axes of the ellipse). Figure 6.4 shows the result of stretching in the y-direction by two. The contour lines of the corresponding plot are now clearly ellipses.

Further anisotropies can be introduced by using correlation functions which do not depend on the distance $x^2 + y^2$ only. For example, we can study the effect of letting the correlation oscillate along the y-direction. This can be achieved, for example, with the following correlation function:

$$\rho(x, y) = \exp\left(-\alpha(x^2 + y^2)\right) \frac{1}{2}\left(1 + \cos(\beta y)\right). \tag{6.2}$$

Figure 6.1: Isotropic Gaussian Correlation Function with $\alpha = 5$

Figure 6.2: Isotropic Gaussian Correlation Function with $\alpha = 0.1$

Figure 6.3: Isotropic Gaussian Correlation Function with $\alpha = 0.01$

Figure 6.4: Ellipsoidal Gaussian Correlation Function with $\alpha = 0.06$

Figure 6.5: Oscillated Gaussian Correlation Function with $\alpha = 0.1$ and $\beta = 4$

Figure 6.6: Oscillated Gaussian Correlation Function with $\alpha = 0.05$ and $\beta = 2$

Figure 6.7: Singular Correlation Function with $a = 0.5$

The effect of parameter α is the same, and the second parameter β determines the size of the oscillations. The influence of these parameters is illustrated by Figures 6.5 and 6.6. We can observe the strong anisotropies from the plots of the correlation function.

The Gaussian is not the only possible correlation function. In general we want functions which decay with distance, and which are nearly zero after some given distance. Another function having these properties is the inverse power function:

$$\rho(x, y) = \frac{1}{1 + (x^2 + y^2)^a}. \tag{6.3}$$

Figures 6.7 and 6.8 show resulting images. For low values of the parameter a, the radius of influence seems to be too big and there are noticeable singularities at the data points. For large values of a the functions drops off too rapidly, and we get a result similar to the image of Figure 6.1.

In all the examples, the phenomenon was constrained to take the same value of 1.0 at the data points. Hence theoretically the "best" interpolator must be the constant function equal to 1.0. This results in a totally blank image! One possible way to force the phenomenon to be zero "far-away" from the data points is to constrain the phenomenon to be zero at a unique data location distant from the data.

6.1.2 Specification of the Small Scale Detail

As stated in Chapter 4 the user has to specify the small scale detail by the second-order statistics of a random function. In most cases these second-order statistics are estimated from actual samples of the phenomenon. However, he or she has the freedom to create any valid correlation or spectral density function. Figure 6.9 shows an image of a phenomenon where small scale has been added using the stochastic Kriging technique described in Section

Figure 6.8: Singular Gaussian Correlation Function with $a = 2.5$

Figure 6.9: Addition of the Small-Scale Detail using Stochastic Kriging

Figure 6.10: 2-D Weierstrass-Mandelbrot function with $M = 4$ and 8

4 of Chapter 4. The small-scale noise for that image is a random function having an isotropic *Markovian* correlation function:

$$\rho(x, y) = \exp\left(-\alpha\sqrt{x^2 + y^2}\right). \tag{6.4}$$

The global shape is constrained by the same data constraints of the previous section, with an isotropic Gaussian correlation function with $\alpha = 0.04$. This small-scale was synthesized using an FFT spectral model described in Section 3.1. The problem with the stochastic Kriging approach is that small-scale detail may also appear far away from the boundary of the global shape of the phenomenon. One solution is to let this small-scale decay with the distance from the global shape. Techniques similar to those described in the simple illumination model of the previous chapter could be used. In the next section more convincing results involving the small-scale detail will be presented.

Several of the random functions discussed in Section 4.3 have been tried. Examples of random samples using Perlin's function can be found in many references in computer graphics (e.g. in [30]). The two-dimensional Weierstrass-Mandelbrot function, however, has not received much attention in computer graphics. Figures 6.10 and 6.11 show four samples of a fractal random field with different values of M. Recall that M is the number of one-dimensional Weierstrass-Mandelbrot functions which are superposed. For the first two images (i.e. $M = 4$ and 8), the directional artifacts (ridges) are clearly visible. Even in the last image with $M = 32$ some artifacts are still visible. It seems that unless one wants these strong directionalities, the multi-dimensional Weierstrass-Mandelbrot function needs too many superpositions to generate artifact-free pictures.

Figure 6.11: 2-D Weierstrass-Mandelbrot function with $M = 16$ and 32

6.2 Three-Dimensional Clouds

In this section we will present results of our model applied to the simulation of three-dimensional clouds. The illumination model used was described in the previous chapter. This illumination model has been directly implemented in a standard ray-tracer; i.e *Optik*, a ray-tracer developed at the University of Toronto by Amanatides and Woo. The global shape is given by the set of data points and coefficients of the Kriging estimator. This supposes that the modelling was done in a prior stage, in other words another program reads the input data supplied by the user and calculates the coefficients of the Kriging estimate and stores them in a format understandable by Optik.

Figures 6.12, 6.13 and 6.14 are images of the global shape with different correlation functions. The user supplied data is represented by small red spheres. To give a depth clue the spheres gradually become faded the farther they are from the viewer. These images are reminiscent of Blinn's blobbies. The isotropic Gaussian correlation function is used. The details of the rendering of these images was covered in Section 5.5.

The addition of small-scale detail using the simple illumination model is illustrated in Figures 6.15, 6.16, 6.17 and 6.18. We used Perlin's function to generate the small-scale detail. This is equivalent to assuming that the small-scale is "fractal-like" (see Section 4.3.3). The noticeable differences in the four images come from different scaling values of Perlin's noise.

We will now experiment with an elliptical correlation function for the global shape model. Figure 6.19 shows an example of the global shape using such a correlation function. The shape of the ellipsoids varies as a function of the height of the cloud. The ellipsoids are "flatter" near the bottom of the cloud. This permits us to simulate the behaviour of cumulus type clouds observed in the sky. Figure 6.20 shows the result of adding the small-scale detail to this model.

Figure 6.12: Image of global shape 1

Figure 6.13: Image of global shape 2

Figure 6.14: Image of global shape 3

Figure 6.15: Addition of small-scale detail 1

Figure 6.16: Addition of small scale detail 2

Figure 6.17: Addition of small-scale detail 3

Figure 6.18: Addition of small-scale detail 4

Figure 6.19: Global shape with elliptical correlation function

Figure 6.20: Addition of small-scale detail to previous elliptical model

In the above images certain artifacts caused by the simplicity of the illumination model are visible. For example, the fuzzy regions at the boundary of the cloud are all very small. Some clouds, however, show big fuzzy "wisps", typically at the tops of clouds. These effects can be modelled by adding more parameters to our simple illumination model, or to use more sophisticated volume rendering approaches based on the algorithm presented in Section 5.3. In both illumination models, however, the availability of the global shape is crucial in the simulation of semi-global illumination effects such as self-shadowing. This means that the user could specify one of the possible rendering techniques necessary to determine the translucence of the cloud at the intersection point. This choice is then a trade-off between image quality and computation time. For example, the simple illumination model could be used to "preview" the scene for modelling purposes.

All pictures have a resolution of 512×512 pixels and were rendered on Iris 4D workstations. The rendering of the global shape took approximately 20 to 30 minutes. With the addition of the small-scale detail rendering times took approximately one hour. Note that there has been no attempt to optimize the code, as clarity and correctness were our first concerns. To solve the Kriging system a simple LR decomposition routine of *linpack* was used.

Chapter 7

Conclusion

The main contribution of this thesis was to introduce a new stochastic model to computer graphics. Unlike previous models, it overcomes the high storage requirements in the case of three-dimensional phenomena without sacrificing user control. This is achieved by considering a model that operates on two or more scales of visual detail.

At the macroscopic level, the shape is given as an interpolator of user supplied data. Since we interpret the interpolation as an estimation problem, the interpolator is controlled by a correlation measure. By necessitating that the estimator is unbiased and has minimal variance, an optimal estimator is calculated. The technique used is called Kriging. This technique requires the solution of one linear system with a size equal to the number of data constraints. This estimator has low storage requirements because it is a weighted sum of the correlation measure at different locations. At the microscopic level, the small-scale detail is modelled by a solid texture. This texture is modelled by a random function which can be evaluated anywhere in 3-space and is given by a small number of coefficients. A good example is Perlin's noise function [30]. Our model has low storage requirements because of the simple form of the estimator and the choice of small-scale random function. In addition our model allows the user to have control over the shape of the phenomenon as a result of the interpolation at the global scale.

The feasibility of our model was demonstrated by applying it to the simulation of clouds. Clouds are modelled as density maps. At the rendering level we took full advantage of the separation of scales. The global shape is used to calculate semi-global illumination effects, such as self-shadowing. We considered two different rendering algorithms to add the translucence. The first algorithm generalized the heuristics first introduced by Gardner [13]. It has the advantage of being simple and suitable for implementation in a standard renderer. The second algorithm is more expensive because it uses volume rendering techniques.

The first algorithm was implemented and the results that were shown in chapter 6 demonstrate that our model is well suited for the rendering of three-dimensional partially-translucent phenomena such as clouds.

The extension of the model from two-scales to more scales of visual detail was mentioned in Chapter 4, however the procedure still remains to be formalized in a rigorous mathematical framework. To achieve this, certain models from low-level vision and image compression could be used. The technique of stochastic Kriging as presented in chapter 4 needs certain modifications to be more useful in computer graphics. The addition of noise should only be concentrated near the global shape and the use of different correlation measures at different scales should be allowed.

The rendering of the model can be improved in different ways. For example, heuristic rendering algorithms other than Gardner's could be designed. As well, although the more expensive volume rendering technique was described in detail in Chapter 5, it remains to be implemented to assess its true advantages.

An interesting area of future research might be to apply the above model to other natural phenomena. Of special interest would be those which have not yet been simulated by previous models because of high storage requirements. The first step would be to apply the model to partially-translucent phenomena different from clouds. An example would be fire. A more ambitious goal would be to try to model water or other fluids using the same model. In particular it would be exciting to incorporate the dimension of time in our random functions. This would permit us to simulate complex dynamic behaviour which is presently beyond purely deterministic models. Furthermore, the temporal correlation function could be used to speed up the rendering of subsequent frames in an animation.

Bibliography

[1] K. A. Anjyo. "A Simple Spectral Approach to Stochastic Modelling for Natural Objects", *Proceedings of EUROGRAPHICS '88* (September 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. A. R. Blais. *Estimation and Spectral Analysis*, The University of Calgary Press, Calgary Alberta, 1988.

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

[6] J. F. Blinn. "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces", *Proceedings of SIGGRAPH '82*, also published as *ACM Computer Graphics, 16, 3* (July 1982), 21-29.

[7] F. L. Bookstein. "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations", *IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, 6* (June 1989), 567-585.

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

[9] P. J. Burt, E. H. Adelson. "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions on Communications, 31, 4* (April 1983), 532-540.

[10] R. F. Cahalan. "Overview of Fractal Clouds", in *RSRM '87: Advances in Remote Sensing Retrieval Methods*, A. Deepak et al., eds. A. Deepak Publishing, 1989.

[11] J. P. Delhomme. "Kriging in the Hydrosciences", *Advances in Water Resources, 1, 5*, 1978, 251-266.

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

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

[14] J. Gleick. *Chaos: Making a New Science*, Penguin Books, New York, 1987.

[15] T. A. Hewett, "Fractal Distributions of Reservoir Heterogeneity and Their Infuluence on Fluid Transport", *SPE 15386, 61st Annual Technical Conference and Exhibition of Petroleum Engineers*, New Orleans, 1986.

[16] T. A. Hewett, R. A. Behrens. "Conditional Simulation of Reservoir Heterogeneity With Fractals", *SPE 18325, 63rd Annual Technical Conference and Exhibition of Petroleum Engineers*, Houston, 1988.

[17] A. G. Journel. "Geostatistics for Conditional Simulation of Orebodies", *Economic Geology, 69, 5*, 1974, 673-687.

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

[19] J. T. Kajiya. "New Techniques for Ray Tracing Procedurally Defined Objects", *Proceedings of SIGGRAPH '83*, also published as *ACM Computer Graphics, 17, 3* (July 1983), 91-102.

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

[21] J. T. Kajiya. "Rendering Fur with Three Dimensional Textures", *Proceedings of SIGGRAPH '89*, also published as *ACM Computer Graphics, 23, 3* (July 1989), 271-280.

[22] J. Lenoble. *Radiative Transfer in Scattering and Absorbing Atmospheres*, A. Deepak Publishing, Hampton, Virginia, 1985.

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

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

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

[26] B. Mandelbrot, J. W. van Ness. "Fractional Brownian Motion, fractional noises and applications", *SIAM Review 10, 4* (1968), 321-340.

[27] J. Meinguet. "Surface Spline Interpolation: Basic Theory and Computational Aspects", in *Approximation Theory and Spline Functions*, S. P. Singh et al., Eds. Dordrecht, The Netherlands: Reidel, 1984, 127-142.

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

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

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

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

[32] R. R. Rogers. *A Short Course in Cloud Physics*, Pergamon Press, Oxford, 1989.

[33] P. Sabella. "A Rendering Algorithm for Visualizing 3D Scalar Fields", *Proceedings of SIGGRAPH '88*, also published as *ACM Computer Graphics, 22, 4* (August 1988), 51-58.

[34] G. Sakas. "Fast Rendering of Arbitrary Distributed Volume Densities", *Proceedings of EUROGRAPHICS '90* (September 1990), North-Holland, 519-530.

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

[36] J. M. Snyder, A. H. Barr, "Ray Tracing Complex Models Containing Surface Tesselations", *Proceedings of SIGGRAPH '87*, also published as *ACM Computer Graphics, 21, 4* (July 1897), 119-128.

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

[38] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-level Vision*, Kluwer Academic Publishers, Norwell MA, 1989.

[39] D. Terzopoulos, J. Platt, A. Barr, K. Fleisher. "Elastically Deformable Models", *Proceedings of SIGGRAPH '87*, also published as *ACM Computer Graphics, 21, 4* (July 1987), 205-214.

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

[41] 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.

[42] L. Williams. "Pyramidal Parametrics", *Proceedings of SIGGRAPH '83*, also published as *ACM Computer Graphics, 17, 3* (July 1983), 1-11.

[43] A. M. Yaglom. *Correlation Theory of Stationary and Related Random Functions I. Basic Results*, Springer-Verlag, 1986.