Stochastic Rendering of Density Fields

Jos Stam

Department of Computer Science
University of Toronto
stam@dgp.toronto.edu

Abstract

Stochastic models are often economical to generate but problematic to render. Most previous algorithms first generate a realization of the stochastic model and then render it. These algorithms become expensive when the realization of the stochastic model is complex, because a large number of primitives have to be rendered. In *stochastic rendering* we also model the intensity as a random field, and the statistics of the intensity field are related to the statistics of the stochastic model through an illumination model. Stochastic rendering algorithms then generate a realization of the intensity field directly from these statistics. In other words, a random component is shifted from the modelling to the rendering. This paradigm is not entirely new in computer graphics, so related work will be discussed. The main contribution of this paper is a stochastic rendering algorithm of gaseous phenomena modelled as random density fields such as clouds, smoke and fire. A simplified version of the scattering equation is used to derive the statistics of the illumination field. Our algorithm is therefore an improvement over similar algorithms both in terms of computational speed and generality.

Résumé

Le rendu de modèles stochastiques est souvent problématique. La plupart des algorithmes existants génèrent d'abord une réalization du modèle stochastique avant le rendu. Ces algorithmes deviennent couteux dans le cas où le modèle est compliqué à cause du grand nombre de primitives qui doivent être rendues. Dans la méthode du *rendu stochastique* nous modélisons l'intensité comme un champs aléatoire, les statistiques du champs d'intensité sont calculées à partir des statistiques du modèle stochastique en utilisant un modèle d'illumination. La méthode du rendu stochastique génère une réalization du champs d'intensité directement à partir de ces statistiques. En d'autres termes la composante stochastique est translatée du modellage au rendu. La contribution majeure de ce papier consiste en un algorithme pour le rendu de phénomènes gazeux tels que les nuages, la fumée et le feu.

Keywords: stochastic modelling, simulation of gaseous phenomena, scattering equation, solid textures, ray tracing.

1 Introduction

The use of stochastic models to capture the complexity of natural phenomena is well established in computer graphics. Phenomena such as terrain, oceans and clouds have all been successfully modelled using this approach. Most research in this area in the past has focused on developing algorithms that generate a realization of the stochastic model. For example, a realization of a fractal terrain is given by a set of connected triangles [4]. In general, the realization is a set of simple geometric primitives which can be rendered using a standard rendering algorithm. The cost of the rendering is therefore directly related to the complexity of the realization. For complex phenomena this means that the rendering can become prohibitively expensive. To avoid this computational explosion we propose a different approach in rendering stochastic models, based on the observation that the intensity field resulting from the interaction of light sources with the stochastic model is random. Therefore, this random intensity field can also be described by a stochastic model. The statistics of the intensity field are related to the statistics of the phenomenon through an illumination equation. Thus, an image of the phenomenon can be computed by generating a realization of the intensity field directly from its statistics. We call this approach *stochastic rendering*. In other words, instead of perturbing the model, we perturb the intensity in a way which is consistent with both the model and the illumination equation.

We shall apply this paradigm to the efficient rendering of gaseous phenomena modelled as random density fields: functions that assign a random density to each spatial location. Clouds, for example, are modelled as a density of water vapour. Previous algorithms generate realizations of the density field by using either procedural models [14] or spectral methods [20]. The realization is then sampled on a three-dimensional lattice and rendered using a volume-tracer. Efficient single-scattering volume tracers have been developed by Ebert [3] and Sakas [16]. The effects of multiple-scattering can be calculated prior to volume tracing at the expense of an increase in computation time. Rushmeier et al. describe a radiosity-based algorithm for density fields having constant scattering properties [15]. Kajiya et al. describe a more general multiple-scattering algorithm based on a decomposition of the intensity field into spherical harmonics [10], however very few details are given about the implementation. For scenes containing complex gaseous phenomena such as a typical sky on a partly cloudy day, these algorithms become very expensive in terms of both computation and storage. Consequently, more efficient algorithms are needed. In this work we describe one solution, a stochastic rendering algorithm for random density fields. To get a tractable algorithm we separate the intensity field into both a direct and indirect component. The statistics of the direct intensity field are calculated using the scattering equation from radiative transfer. We use the fact that indirect intensity usually has low variance and can therefore be calculated using only average properties of the density field. Under these conditions an efficient rendering algorithm can be derived.

This paper will be divided into eight sections. In Section 2 we will review the basic theory of random fields in order to precisely define the statement "statistical description". Section 3 describes the stochastic rendering paradigm in more detail and refers to related work. In Section 4 we will delineate the statistical model used to model random density fields. In Section 5 we will briefly state the scattering equation and specify our simplifying assumptions. Section 6 explicates our new algorithm. In Section 7 we will summarize results obtained. Conclusions and further work will be discussed in Section 8.

2 Random Fields

In this section we precisely explain the notions of random field and statistical description. A random field $R(\mathbf{x})$ is a function that assigns a random variable for each spatial position $\mathbf{x} = (x, y, z)$. Particular instances of the random variables define a (real) function known as a *realization* of the random field. A random field is entirely defined if the sequence of *distribution probabilities* F_n $(n \geq 1)$ of its realizations is specified:

$$F_n(r_1, \mathbf{x}_1; \ldots; r_n, \mathbf{x}_n) = \mathrm{Prob}(R(\mathbf{x}_i) \leq r_i, i = 1, \ldots, n).$$

In practice, however, there is neither a way to find this sequence, even for finite n, nor a need for this since a random field can be described by its *statistical moments*. These moments are obtained by averaging over many realizations $R^{(1)}, \ldots, R^{(N)}$:

$$\langle R(\mathbf{x}) \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} R^{(n)}(\mathbf{x}).$$

The most important statistical moments of a random field are its *mean* $\overline{R}$ and its *covariance* K_R given by

$$\begin{aligned} \overline{R}(\mathbf{x}) &= \langle R(\mathbf{x}) \rangle \quad \text{and} \\ K_R(\mathbf{x}'; \mathbf{x}'') &= \langle R(\mathbf{x}') R(\mathbf{x}'') \rangle - \overline{R}(\mathbf{x}') \overline{R}(\mathbf{x}''). \end{aligned}$$

The mean is the average value at each point and the covariance is the amount of correlation between two points. The spread around the mean is given by the *variance* $\sigma_R^2(\mathbf{x}) = K_R(\mathbf{x}; \mathbf{x})$. In practice, the mean and the covariance are enough to fully characterize a random field. It is customary in computer graphics [13] to assume that higher-order moments are irrelevant for visual models. In this paper a *statistical description* refers to the specification of these two moments. In case the covariance depends only on the difference between two points, the random field is *homogeneous*. Homogeneous random fields are useful because they have a representation in the frequency domain. This is desirable since efficient algorithms based on the fast fourier transform exist to generate such random fields [20]. The drawback, however, is that homogeneous random fields have a constant mean and variance. Many phenomena that exhibit large scale variations cannot therefore be modelled by homogeneous random fields. In this work we consider non-homogeneous random fields that are simple transformations of homogeneous random fields, specifically random fields of the form

$$R(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) Q(\mathbf{x}), \tag{1}$$

where Q is a homogeneous random field with zero mean and covariance $K_Q(\mathbf{x}'; \mathbf{x}'') = K_Q(\mathbf{x}'' - \mathbf{x}'; \mathbf{0})$. The deterministic functions f and g are used to model a desired mean and variance. A simple calculation shows that the mean and covariance of the transformed field are equal to

$$\overline{R}(\mathbf{x}) = f(\mathbf{x}) \text{ and } K_R(\mathbf{x}'; \mathbf{x}'') = K_Q(\mathbf{x}'' - \mathbf{x}') g(\mathbf{x}') g(\mathbf{x}''). \tag{2}$$

In particular the variance is given by $\sigma_R^2(\mathbf{x}) = g^2(\mathbf{x})$. A random field with an arbitrary mean and variance can therefore be modelled by choosing the functions f and g accordingly. We have chosen this particular transformation as it gives us total control over both the mean and variance. We will now discuss the use of random fields in rendering. The temporal behaviour of the random field can be modelled by allowing the functions f and g to vary over time and by specifying the temporal statistics of the homogeneous perturbation Q.

3 Stochastic Rendering

In general, the intensity of light at a given point is a function of the geometry of the scene, the initial lighting conditions (light sources), the reflective properties of surfaces and the scattering properties of a participating medium [9]. If any of these parts is random, then the resulting intensity field will also be random. The statistical description of the intensity field is a function of the statistical descriptions of the parts (via the illumination model used). In cases when this statistical description can be established, a realization of the random intensity field can be computed. In many instances this will result in an efficient rendering algorithm. To our knowledge no work has been devoted in computer graphics to calculating these statistics, except for the derivation of reflectance models for surfaces. Following is a brief discussion of these models.

Many reflectance models for surfaces are derived using a statistical description of the surface. Usually only the mean value of the intensity reflected from the surface is calculated. Torrance and Sparrow model the surface by a random distribution of normals [19]. Their model has been used in many local illumination models. The fluctuations around the mean are usually accounted for by adding an arbitrary value through a texture map or by perturbing the mean normal of the surface [1]. Recently He et al. generalized the work of Torrance and Sparrow to a wave-physics description of light [7]. The only work in computer graphics which addresses the problem of calculating the covariance of the reflected intensity field to our knowledge is the work of Krueger [12]. His calculations become tractable for certain idealized situations (e.g., planar surfaces). The inclusion of the covariance in a reflectance model remains, however, an unexplored area of research in computer graphics. This is due mainly to the mathematical complexity of the task.

In the case of volume rendering, attempts to calculate the statistical description of the intensity field have not been published. However, Gardner's algorithm to render clouds has the semblance of a stochastic rendering algorithm in that he perturbs the transparency rather than the density of the clouds [5]. Therefore randomness is added in the rendering instead of in the modelling as in our algorithm. His model is heuristic and is not based on any well established equation that describes the transfer of light in density fields. Instead Gardner's model is a modification of a standard illumination model for surfaces. Because his algorithm is surface based it has some shortcomings. For example, objects cannot disappear smoothly through his clouds. Recently Kaneda et al. have used Gardner's model in conjunction with their atmospheric illumination model [11]. They have rendered the most realistic pictures of clouds to date.

In this paper we explore only a small set of the possible applications of stochastic rendering. For example, algorithms could be devised to deal with random light sources.

4 Statistical Description of the Density Field

Complex gaseous phenomena are conveniently modelled by a random density field, a function which assigns a random density variable $\rho(\mathbf{x})$ to each spatial location $\mathbf{x}$. In order to define the random density field, both its mean and covariance must be specified.

The mean describes the large scale variations ("global shape") of the density field. A natural model for the mean is therefore a smooth function. Many choices are possible, from splines to interpolation schemes. In this work we choose to model the mean as a smooth blending of scattered data values, so that the model can be used both in design and visualization. In design the data is provided by a user, thus allowing control over the global features of the density [17]. In visualization the data is sampled using a measuring device. More formally, given values for the masses m_i at n points $\mathbf{x}_i$, the mean is given by:

$$\overline{\rho}(\mathbf{x}) = \sum_{i=1}^{n} m_i W(\mathbf{x} - \mathbf{x}_i) = \sum_{i=1}^{n} \overline{\rho}_i(\mathbf{x}), \tag{3}$$

where W is a smooth blending function satisfying $\int W(\mathbf{x})\, d\mathbf{x} = 1$. A similar mean is obtained if we model the density as a sum of n random fields ρ_i with each having a mean $\overline{\rho}_i$. In other words, the random field can be seen as a weighted sum of "random blobs". For our purposes this interpretation is very convenient. Consequently, we describe the covariance functions of each blob instead of the covariance of the entire density field. We want the variance of each blob to be proportional to its mean. A natural choice then, is to make the variance directly proportional to the mean: $\sigma^2_{\rho,i} = \epsilon\overline{\rho}_i$. With this choice, variations tend to become too large near the centre of the blob which is unrealistic. We expect the variance to drop off with the mean but to not grow to exceedingly large values. We can avoid the latter by "clamping" the values of the mean: $\sigma^2_{\rho,i} = \epsilon F_\eta(\overline{\rho}_i)$. The clamping function is defined as:

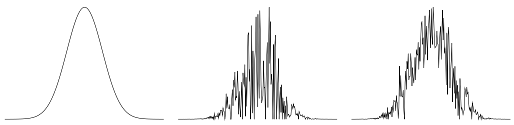

Figure 1: $\sigma^2_{\rho,i} = 0$, $\sigma^2_{\rho,i} = \epsilon\overline{\rho}_i$ and $\sigma^2_{\rho,i} = \epsilon F_\eta(\overline{\rho}_i)$

$$F_\eta(t) = \begin{cases} t & \text{if } t < \eta \\ \eta & \text{otherwise.} \end{cases}$$

Figure 1 shows the effect of the clamping function on the resulting random blob. The cutoff parameter η and the magnitude ϵ are either provided by a user or are estimated from data. We assume that the random blobs are transformations of a single homogeneous random field Q (see Eq. 1): $\rho_i = \overline{\rho}_i + \sigma_{\rho,i} Q$. The covariance of each blob therefore has the following form:

$$K_{\rho,i}(\mathbf{x}';\mathbf{x}'') = K_Q(\mathbf{x}'' - \mathbf{x}')\sigma_{\rho,i}(\mathbf{x}')\sigma_{\rho,i}(\mathbf{x}''). \tag{4}$$

We assume that the covariance of the homogeneous random field Q is a gaussian with standard deviation α:[1]

$$K_Q(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{3}{2}}\,\alpha^3} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\alpha^2}\right). \tag{5}$$

Realizations of homogeneous random fields with a gaussian covariance are easy to generate, because the fourier transform of a gaussian is well defined [20].

5 Illumination Model

In this section we describe the illumination model that we use to derive the illumination field statistics. We begin with a brief statement of the scattering equation.

5.1 The Scattering Equation

A density field alters the intensity of light using three effects: absorption, scattering and emission. To describe how these effects alter the intensity of light along a ray, consider a ray $\mathbf{x}_u = \mathbf{x}_0 - u\mathbf{s}$ with origin $\mathbf{x}_0$ and direction $-\mathbf{s}$. The light at point $\mathbf{x}_0$ shining in direction $\mathbf{s}$ is the sum of two contributions [8]:[2]

$$I(\mathbf{x}_0,\mathbf{s}) = \tau(0,b)I(\mathbf{x}_b,\mathbf{s}) + \int_0^b \tau(0,u)\rho(\mathbf{x}_u)\kappa_t J(\mathbf{x}_u,\mathbf{s})\, du. \tag{6}$$

The first term is the proportion of light emitted at point $\mathbf{x}_b$ that is not absorbed or scattered in other directions

[1] This assumption is significant in Section 5.2 of this paper.

[2] All functions in this section are understood to be wavelength dependent.

by the density field. The attenuation factor τ is called the *transparency* and is given by

$$\tau(u_1, u_2) = \exp\left(-\kappa_t \int_{u_1}^{u_2} \rho(\mathbf{x}_u)\, du\right), \qquad (7)$$

where κ_t is quantity characterizing the amount of light that is absorbed or scattered per unit length. By definition, the transparency has values between zero (total opacity) and one (total transparency). The second term accounts for the intensity of light due to the density field. The function J, the *source function*, describes the increase in intensity at each point. This is due to both the emission and scattering of light from other directions into the direction of the ray:

$$J(\mathbf{x}, \mathrm{s}) = (1-\Omega)E(\mathbf{x}) + \frac{\Omega}{4\pi}\int_{4\pi} p(\mathrm{s}, \mathrm{s}')I(\mathbf{x}, \mathrm{s}')\, d\omega'.$$

The function E denotes the emission of intensity by the density, such as in the case of a fire. This is usually proportional to the density. The second term on the right hand side is an integration of all incoming intensity from all possible solid angles. The function p is called the *phase function* and models the spherical distribution of scattered light. The factor $\Omega \in [0,1]$ known as the *albedo* gives the fraction of light that is scattered. We can rewrite Eq. 6 in terms of the transparency only by defining the *average source function* along the ray by:

$$\begin{aligned}\overline{J}(\mathbf{x}_0, \mathrm{s}) &= \frac{\int_0^b \tau(0,u)\rho(\mathbf{x}_u)\kappa_t J(\mathbf{x}_u, \mathrm{s})\, du}{\int_0^b \tau(0,u)\rho(\mathbf{x}_u)\kappa_t\, du} \\ &= \frac{\int_0^b \tau(0,u)\rho(\mathbf{x}_u)\kappa_t J(\mathbf{x}_u, \mathrm{s})\, du}{1-\tau(0,b)}. \qquad (8)\end{aligned}$$

With this definition the intensity of light reaching point $\mathbf{x}_0$ along the ray can be expressed as a linear combination of the background illumination $I(\mathbf{x}_b, \mathrm{s})$ and the average source function along the ray:

$$I(\mathbf{x}_0, \mathrm{s}) = \tau(0,b)I(\mathbf{x}_b, \mathrm{s}) + (1-\tau(0,b))\overline{J}(\mathbf{x}_0, \mathrm{s}). \qquad (9)$$

Following we describe the statistics of this equation.

5.2 Statistical Equations

Because the density field is random, the transparency and the average source function in Eq. 9 are also random. To derive the statistics of the transparency, we make the assumption that all rays originate from the xy plane and that their directions are in the positive z direction: $\mathbf{x}_u = (x_0, y_0, u)$. In other words, we assume an orthographic projection. The integral in the equation of the transparency (Eq. 7) then becomes a function of the location (x_0, y_0) in the xy plane:

$$\begin{aligned}T(x_0, y_0) &= \int_0^b \rho(x_0, y_0, u)\, du = \sum_{i=1}^n \int_0^b \rho_i(x_0, y_0, u)\, du \\ &= \sum_{i=1}^n T_i(x_0, y_0).\end{aligned}$$

Each integral T_i is a random field which has a well defined mean and covariance [6]:

$$\overline{T}_i(x_0, y_0) = \int_0^b \overline{\rho}_i(x_0, y_0, u)\, du,$$

$$K_{T,i}(x_0, y_0; x_1, y_1) = \int_0^b\int_0^{b'} K_{\rho,i}(x_0, y_0, u; x_1, y_1, u')\, du\, du'. \qquad (10)$$

In other words, the mean and covariance of T_i are directly related to the mean and covariance of the density ρ_i through an integral. Using the fact that the covariance of the homogeneous random field Q is gaussian, the covariance of the integral T_i becomes (see Eq. 4):[3]

$$\begin{aligned}&K_{T,i}(x_0, y_0; x_1, y_1) = K_Q(x_1-x_0, y_1-y_0, 0) \times \\ &\int_0^b\int_0^{b'} K_Q(0,0,u'-u)\sigma_{\rho,i}(x_0, y_0, u)\sigma_{\rho,i}(x_1, y_1, u')\, du\, du' \\ &\approx K_Q(x_1-x_0, y_1-y_0, 0)\int_0^b \sigma_{\rho,i}(x_0, y_0, u)\sigma_{\rho,i}(x_1, y_1, u)\, du. \qquad (11)\end{aligned}$$

Eq. 11 depends on the fact that the support of the covariance K_Q is much smaller than the support of the variance $\sigma^2_{\rho,i}$:

$$\int_0^{b'} K_Q(0,0,u'-u)\sigma_{\rho,i}(x_1, y_1, u')\, du' \approx \sigma_{\rho,i}(x_1, y_1, u).$$

In the limiting case where the covariance is a delta function the above relation is exact. Equation 11 demonstrates that the spatial structure of each integral T_i is essentially the same as the spatial structure of a "slice" of the density field. They differ in that they have a different mean and variance. In particular the variance is given by:

$$\sigma^2_{T,i}(x_0, y_0) = K_Q(0,0,0)\epsilon\int_0^b F_\eta(\overline{\rho}_i(x_0, y_0, u))\, du. \qquad (12)$$

In case both the mean and the covariance are computable, then a realization $T_i^{(0)}$ for each of the integrals can be generated:

$$T_i^{(0)}(x_0, y_0) = \overline{T}_i(x_0, y_0) + \sigma_{T,i}(x_0, y_0)Q^{(0)}(x_0, y_0, 0), \qquad (13)$$

where $Q^{(0)}$ is a realization of the homogeneous random field. A realization for the transparency is then equal to $\exp(-\kappa_t \sum_{i=1}^n T_i^{(0)}(x_0, y_0))$. In the next section we will describe an efficient computation of the mean and the variance.

The statistics of the average source function are tremendously more complicated to calculate. In this work we will assume that the average source function is only a function of the average of the density field. There are good justifications for this assumption. First, the average source function is in fact a convolution of the source function by the "weighting" function $\tau(0,u)\rho(\mathbf{x}_u)\kappa_t$ along the ray. Second, the source function itself involves an integral accounting for scattering

[3] We use the multiplicative property of a gaussian: $K_Q(x,y,z) = K_Q(x,0,0)K_Q(0,y,0)K_Q(0,0,z)$.

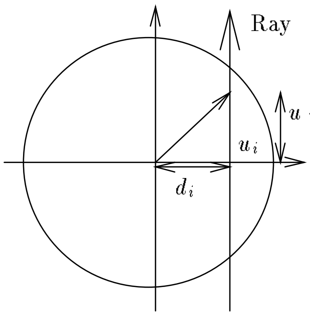

Figure 2: Geometry of Ray and Blob

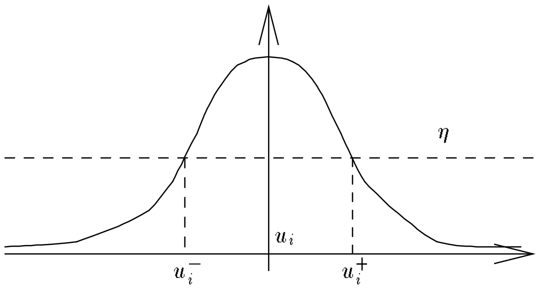

Figure 3: Clamping of the Blob

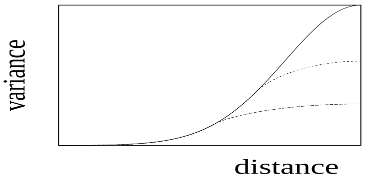

Figure 4: Variance $\sigma_{T,i}^2$ versus the distance d_i for different values of the clamping value η

from different directions. This scattering acts to smooth the intensity field. In fact it is sometimes modelled as a diffusion process [8]. The same smoothing phenomenon due to indirect illumination is also observed in radiosity environments [2].

6 Implementation

In this section we give an efficient algorithm to calculate both the mean and the variance of the integrals in the transparency. We also give an algorithm to calculate the average source function from the mean density field.

6.1 Calculation of the Transparency

We assume that the smoothing kernel that defines the mean density (see Eq. 3) depends only on the magnitude of its argument $W(\mathbf{x}) = W(\|\mathbf{x}\|)$. In this case the integral of the mean of each blob only depends on the distance d_i of the ray to the centre of the blob. Refer to Figure 2 for the geometry of the situation. Let u_i be the parameter value of the point on the ray closest to the centre of the blob. The integral is then given by:

$$\begin{aligned} \overline{T}_i &= m_i \int_0^b W\left(\sqrt{d_i^2 + (u-u_i)^2}\right)\, du \\ &= m_i \left(\mathrm{Table}[d_i, b-u_i] - \mathrm{Table}[d_i, 0-u_i]\right). \end{aligned}$$

The table is computed for discrete values prior to rendering:

$$\mathrm{Table}[d, x] = \int_0^x W\left(\sqrt{d^2 + u^2}\right)\, du.$$

The integral can then be evaluated efficiently by interpolating entries in this table. In order to calculate the variance, a clamped version of the mean density must be integrated (see Eq. 12). Let u_i^- and u_i^+ be the two parameter values on the ray between which the mean density of the blob is clamped, i.e., for which $m_i W(\|\mathbf{x}_u - \mathbf{x}_i\|) = \eta$ (see Figure 3). These values may not be defined in case the maximum of the mean is smaller than the clamping value, in which case we set $u_i^- = u_i^+ = u_i$. The integral can be computed by integrating over each interval separately:

$$\int_0^b F_\eta(\overline{\rho}_i(\mathbf{x}_u))\, du = \int_0^{u_i^-} \overline{\rho}_i(\mathbf{x}_u)\, du + \int_{u_i^-}^{u_i^+} \eta\, du + \int_{u_i^+}^b \overline{\rho}_i(\mathbf{x}_u)\, du =$$
$$m_i \left(\mathrm{Table}[d_i, u_i^- - u_i] - \mathrm{Table}[d_i, 0-u_i] + (u_i^+ - u_i^-)\eta + \mathrm{Table}[d_i, b-u_i] - \mathrm{Table}[d_i, u^+ - u_i]\right).$$

In Appendix A we give complete equations for the case where the smoothing kernel W is gaussian. Realizations of the integrals T_i can therefore be generated efficiently using Eq. 13: $T_i^{(0)} = \overline{T}_i + \sigma_{T,i} Q^{(0)}(\mathbf{x}_{u_i})$. The choice of $\mathbf{x}_{u_i}$ (the point closest to the centre of the blob) guarantees that the perturbation is sampled on a plane for each frame and that the samples are chosen coherently from frame to frame. The latter is important in animated sequences in which the viewing conditions change. In Figure 4 we show plots of the variance $\sigma_{T,i}^2$ versus the distance to the centre of the blob d_i for different values of the clamping value η. As expected, the growth of the variance becomes smaller for small values of the distance, i.e., when the ray is near the centre of the blob. The plain line indicates no clamping and is equal to the mean $\overline{T}_i$. It is interesting to note that Gardner arbitrarily varied the threshold of his transparency function in a similar way. To derive the statistical description of the integrals T_i, we have assumed that the viewing rays are parallel. This is of course not the case in most practical situations. However, if the size of each blob is small with respect to the image, then the rays emanating from a point are approximately parallel across each blob.

6.2 Calculation of the Average Source Function

As we stated in Section 5.2 we assume that the average source function $\overline{J}(\mathbf{x}_0, \mathbf{s})$ (see Eq. 8) depends only on the mean density. We can therefore directly use the rendering algorithm described in [18] which renders density distributions modelled as a weighted sum of blobs of the type given in Eq. 3. The rendering algorithm assumes single-scattering. We refer the reader to reference [18] for a description of the algorithm.

6.3 Overview of the Algorithm

Following we give an overview of our stochastic rendering algorithm. This algorithm can be implemented directly into a standard ray tracer. For each ray, the intensity of light $I(\mathbf{x}_b, \mathbf{s})$ in absence of a density field is calculated using a standard ray-tracer. This illumination is then attenuated and the light created within the density field is added to it as follows:

> `T = 0`
> `I_b =` $I(\mathbf{x}_b, \mathbf{s})$
> `for` each blob intersecting the ray `do`
>   Calculate `av_T =` $\overline{T}_i$ and `sigma_T =` $\sigma_{T,i}$
>   Sample Q at midpoint: `Q =` $Q^{(0)}(\mathbf{x}_{u_i})$
>   Update integral: `T = T + av_T + sigma_T*Q`
> `end for`
> Calculate transparency: `tau = exp(-`κ_t`*T)`
> Calculate `av_J =` $\overline{J}(\mathbf{x}_0, \mathbf{s})$ from mean density field using the algorithm described in [18]
> Combine: `I = tau*I_b + (1-tau)*av_J`

In cases when a shadow ray is computed, only the transparency `tau` must be calculated. The complexity of this algorithm is $O(n)$, where n is the number of blobs. Unlike volume rendering algorithms, our algorithm is therefore independent of the resolution of the random perturbation of the density field. On average our algorithm can be improved by an order of magnitude by grouping the blobs together into a hierarchical data structure of bounding volumes [18]. In the next section we describe results obtained using our algorithm.

7 Results

We implemented the stochastic rendering algorithm into a standard ray-tracer. Prior to rendering, we precompute the tables for the transparency and we generate a realization of the homogeneous random field Q. The latter is generated using an inverse spectral method [20]. This method has the advantage that the resulting realization can be defined and is periodic on a regular three-dimensional lattice. Therefore it can be evaluated efficiently using trilinear interpolation and is defined for all points in space. The artifacts caused by the periodicity of the field are not visible in our images because the field is used only to perturb the illumination. We used a table of size $32 \times 32 \times 32$ in all of our results. A user has control over the shape of the density field by specifying the position, centre and mass of each blob of the mean density (see Section 4). The perturbation is controlled by providing the magnitude ϵ of the variance, the clamping value η and the covariance function of the homogeneous field Q (standard deviation α, see Eq. 5). The illumination properties of the density field are defined by the following parameters: the extinction coefficient κ_t, the albedo Ω and the emission function E (see Section 5.1). In all the results we have assumed a constant phase function p, i.e., scattering is constant in all directions. Next we specify some of our results obtained using the stochastic rendering algorithm.

Clouds. We model clouds in a way similar to Gardner. An individual cloud is generated by randomly placing blobs in an ellipsoid provided by a user. The size and mass of each blob are made inversely proportional to their distance from the centre of the ellipsoid to ensure that the density is maximum in the centre of the cloud. Clusters of individual clouds can be generated by randomly generating such ellipsoids. In Figure 5 we show four pictures of the same cloud with different values for the magnitude ϵ and the clamping value η. Clouds in the left column have a lower perturbation than the ones on the right; and the ones on the bottom have a lower threshold value than the ones on the top. Figure 6 shows one of the clouds with self shadowing, two different clusterings of such clouds and an areal view of downtown on a foggy day. Clouds are characterized by a high amount of scattering, therefore we have set the albedo close to one and have used a non zero value for the emission function, in an attempt to model multiple-scattering. Figure 7 shows four frames from an animated sequence of a cloud interacting with Toronto's CN Tower. This demonstrates that our model handles the interaction of clouds with solid objects. This is an improvement over Gardner's illumination model.

Fire. We have used our algorithm to simulate fire. The flame in Figure 8 was modelled using 14 blobs with different emission parameters. The emission values drop off as a function of the distance of the flame. We have also changed the colour of the emission for higher realism. The smoke on the top of the flame has zero emission and is therefore a perfect absorber. The albedo was set to zero. The illumination of the flame onto the walls and the floor was modelled by placing 5 point light sources within the flame. Note the random patterns generated by the sources.

The rendering times are of the same order of magnitude as the computation of the average source function [18]: one to ten minutes for a typical frame (640×480) on an Iris Indigo.

8 Conclusions and Further Work

In this paper we have presented a new algorithm to render random density fields. Our algorithm is based on a new paradigm called stochastic rendering. Instead of generating the random density field and then using volume rendering, we derive the statistics of the intensity field from the statistics of the density field. From this statistical description we then generate a realization of the intensity field directly. This has the advantage that only a few random samples have to be evaluated, versus the many samples needed in the case of volume rendering. Our results demonstrate that many interesting density distributions can be modelled and rendered efficiently using our algorithm. Our algorithm is also an improvement over similar algorithms since it does not have the artifacts and drawbacks of Gardner's surface-based model. However, we needed to make certain simplifying assumptions in order to get a tractable algorithm. Our algorithm therefore is not as general as the more sophisticated volume renderers. Improvements of our algorithm will be the focus of future research. This includes: modelling of multiple-scattering, deriving the statistics for a perspective projection and calculating the statistical description of the average source function. Generally speaking, the calculation of the statistics of an intensity field is a difficult task. The lack of such descriptions for surface reflectance models demonstrates this. There are still many other applications of the stochastic rendering paradigm which have not yet been explored. For example, the illumination from flickering light sources has not been studied to our knowledge.

9 Acknowledgments

Thanks to Eugene Fiume for supervising this work. Thanks to Pamela Jackson for correcting and improving the writing of this paper. Thanks to the Natural Sciences and Engineering Council of Canada, the Ontario Information Technology Research Centre, and the University of Toronto for their financial support.

A Integrals for a Gaussian Smoothing Kernel

Let W be a gaussian with standard deviation h:

$$W_h(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{3}{2}}\, h^3} \exp\left(-\frac{\|\mathbf{x}\|^2}{2h^2} \right).$$

The mean integral $\overline{T}_i$ for each blob is given by:

$$\begin{aligned} \overline{T}_i &= \int_0^b \frac{m_i}{(2\pi)^{\frac{3}{2}}\, h^3} \exp\left(-\frac{d_i^2 + (u-u_i)^2}{2h^2} \right)\, du \\ &= \frac{m_i}{(2\pi)^{\frac{3}{2}}\, h^2} \exp\left(-\frac{d_i^2}{2h^2} \right) \left(\Phi\left(\frac{b-u_i}{h}\right) - \Phi\left(\frac{0-u_i}{h}\right) \right). \end{aligned}$$

In this case the table is one-dimensional because of the multiplicative properties of the gaussian. The function Φ is defined as:

$$\Phi(u) = \int_0^u \exp\left(-\frac{t^2}{2} \right)\, dt.$$

The values of u_i^- and u_i^+ for which the gaussian is equal to the clamping factor η are given by $u_i^{\pm} = u_i \pm \Delta u_i$, where:

$$\Delta u_i = \sqrt{2 \log \left(\frac{m_i}{(2\pi)^{\frac{3}{2}}\, h^3 \eta} \exp\left(-\frac{d_i^2}{2h^2} \right) \right)}.$$

References

[1] J. F. Blinn. "Simulation of Wrinkled Surfaces". *ACM Computer Graphics (SIGGRAPH '78)*, 12(3):286–292, 1978.

[2] S. E. Chen, H. E. Rushmeier, G. Miller, and D. Turner. "A Progressive Multi-Pass Method for Global Illumination". *ACM Computer Graphics (SIGGRAPH '91)*, 25(4):165–174, July 1991.

[3] D. S. Ebert and R. E. Parent. "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques". *ACM Computer Graphics (SIGGRAPH '90)*, 24(4):357–366, August 1990.

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

[5] G. Y. Gardner. "Visual Simulation of Clouds". *ACM Computer Graphics (SIGGRAPH '85)*, 19(3):297–384, July 1985.

[6] W. A. Gardner. *Introduction to Random Processes.* MacMillan Publishing Company, New York, 1986.

[7] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. "A Comprehensive Physical Model for Light Reflection". *ACM Computer Graphics (SIGGRAPH '91)*, 25(4):175–186, July 1991.

[8] A. Ishimaru. *VOLUME 1. Wave Propagation and Scattering in Random Media. Single Scattering and Transport Theory.* Academic Press, New York, 1978.

[9] J. T. Kajiya. "The Rendering Equation". *ACM Computer Graphics (SIGGRAPH '86)*, 20(4):143–150, August 1986.

[10] J. T. Kajiya and B. P. von Herzen. "Ray Tracing Volume Densities". *ACM Computer Graphics (SIGGRAPH '84)*, 18(3):165–174, July 1984.

[11] K. Kaneda, T. Okamoto, E. Nakamae, and T. Nishita. "Photorealistic Image Synthesis for Outdoor Scenery under Various Atmospheric Conditions". *The Visual Computer*, 7:247–258, 1991.

[12] W. Krueger. "Intensity Fluctuations and Natural Texturing". *ACM Computer Graphics (SIGGRAPH '88)*, 22(4):213–220, August 1988.

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

[14] K. Perlin. "An Image Synthesizer". *ACM Computer Graphics (SIGGRAPH '85)*, 19(3):287–296, July 1985.

[15] H. E. Rushmeier and K. E. Torrance. "The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium". *ACM Computer Graphics (SIGGRAPH '87)*, 21(4):293–302, July 1987.

[16] G. Sakas. "Fast Rendering of Arbitrary Distributed Volume Densities". In F. H. Post and W. Barth, editors, *Proceedings of EUROGRAPHICS '90*, pages 519–530. Elsevier Science Publishers B.V. (North-Holland), September 1990.

[17] J. Stam and E. Fiume. " A Multiple-Scale Stochastic Modelling Primitive". In *Proceedings of Graphics Interface '91*, pages 24–31, June 1991.

[18] J. Stam and E. Fiume. "Turbulent Wind Fields for Gaseous Phenomena". In *Proceedings of SIGGRAPH '93*, pages 369–376. Addison-Wesley Publishing Company, August 1993.

[19] K. E. Torrance and E. M. Sparrow. "Theory for Off-Specular Reflection From Roughened Surfaces". *Journal of the Optical Society of America*, 57(9):1105–1114, September 1967.

[20] R. P. Voss. "Fractal Forgeries". In R. A. Earnshaw, editor, *Fundamental Algorithms for Computer Graphics.* Springer-Verlag, 1985.

Figure 5:

Figure 7:

Figure 6:

Figure 8: