[6] A. Ishimaru, *Wave Propagation and Scattering in Random Media*, Academic Press, 1978.

[7] J.T. Kajiya, *The Rendering Equation*, Proc. SIGGRAPH'86, Computer Graphics, v20, n3, p143-145, 1986.

[8] J.T. Kajiya, B. Von Hertzen, *Ray Tracing Volume Densities*, Proc. SIGGRAPH'84, Computer Graphics, v18, n3, p165-174, 1984.

[9] B. Le Saëc, C. Schlick, *A Progressive Ray Tracing based Radiosity with General Reflectance Functions*, Proc. of First Eurographics Workshop on Rendering (Rennes), p103-116, 1990.

[10] N.L. Max, *Atmospheric Illumination and Shadows*, Proc. SIGGRAPH'86, Computer Graphics, v20, n4, p117-124, 1987.

[11] G.W. Meyer, *Wavelength Selection for Synthetic Image Generation*, Computer Graphics & Image Processing, n41, p57-79, 1988.

[12] T. Nishita, Y. Miyawaki, E. Nakamae, *A Shading Model for Atmospheric Scattering considering Luminous Distribution of Light Sources*, Proc. SIGGRAPH'87, Computer Graphics, v21, n4, p303-310, 1987.

[13] H.O. Peitgen, D. Saupe, *The Science of Fractal Images*, Springer Verlag, 1988.

[14] K. Perlin, *An Image Synthesizer*, Proc. SIGGRAPH'85, Computer Graphics, v19, n3, p287-296, 1985.

[15] K. Perlin, E.M. Hoffer *Hypertexture*, Proc. SIGGRAPH'89, Computer Graphics, v23, n3, p253-267, 1989.

[16] H.E. Rushmeier, K.E. Torrance, *The Zonal Method for calculating Light Intensities in Presence of Participating Medium*, Proc. SIGGRAPH'87, Computer Graphics, v21, n4, p293-302, 1987.

[17] G. Sakas, *Fast Rendering of Arbitrary Distributed Volume Densities*, Proc. EUROGRAPHICS'90, p519-530, 1990.

[18] C. Schlick, *Divers éléments pour une synthèse d'images réalistes*, PhD Thesis, Université Bordeaux 1, November 1992.

[19] J.R. Wallace, K.A. Elmquist, E.A. Haines *A Ray Tracing Algorithm for Progressive Radiosity* , Proc. SIGGRAPH'89, Computer Graphics, v23, n3, p315-324, 1989.

While $A >$ THRESHOLD and R is still in O

Add the attenuated radiance of V : $L_P = L_P + AL_V$.

Scale the attenuation $A = Ae^{-\alpha\rho\delta}$ where δ = distance covered in V and ρ = density of V.

End-While.

If $A >$ THRESHOLD Add the attenuated radiance of the background : $L_P = L_P + AL_\infty$.

End-For.

5 Results

Picture 1, 2 and 3 show different views of a volume density object simulating a cloud illuminated by a single source. The cloud is defined by a $65 \times 65 \times 10$ grid. The pictures were computed on a Sun SparcStation and visualized on a Silicon Graphics Personal Iris at a 400×400 resolution. For the first pass, 2 million rays have been sent and the CPU time ranged from 3 to 4 minutes. For the second pass, 1.6 million rays have been sent and the CPU time ranged from 1 to 2 minutes.

6 Conclusion

We have presented a new algorithm for simulating the effect of light travelling through volume density objects (haze, fog, clouds...) modelized by voxel grids which define their density distribution in a discrete tridimensional space. The method is a two-pass Monte-Carlo ray-tracing including the following features :

- No restrictive assumptions are made, neither about the characteristics of the objects (both arbitrary density distributions and phase functions are allowed) nor about the physical phenomena included in the rendering process (multiple scattering is accounted for).
- The phase function is used to express the deviation of light at a scattering point rather than the ratio of energy between two directions.
- A new familly of phase functions is proposed that provides a continuum from perfect backward scattering to perfect forward scattering, and is particularly well-suited for efficient sampling in the Monte-Carlo process that generates the new direction of a ray after scattering.
- Although it completely deals with multiple scattering, the method is very inexpensive compared to equivalent techniques [8, 16] because the use of separated Markov chains [7] in the Monte-Carlo process prevents the number of rays to grow exponentially.

The current implementation treats only direct illuminations of the volume density objects, but it can be easily integrated in a global illumination model using a progressive ray-tracing based radiosity algorithm like those proposed in [9, 19]. Such an integration is in progress by our team at LaBRI.

References

[1] J.F. Blinn, *Light Reflection Functions for Simulation of Clouds and Dusty Surfaces*, Proc. SIGGRAPH'82, Computer Graphics, v16, n3, p21-29, 1982.

[2] D.S. Ebert, R.E.Parent, *Rendering and Animation of Gaseous Phenomena*, Proc. SIGGRAPH'90, Computer Graphics, v24, n4, p357-366, 1990.

[3] J.M. Hammerley, D.C. Handscomb, *Monte-Carlo methods*, Wiley, 1964.

[4] S. Haas, G. Sakas, *Methods for effficient Sampling of Arbitrary Distributed Volume Densities*, Proc. of First Eurographics Workshop on Rendering (Rennes), p215-227, 1990.

[5] M. Inakage, *Volume Tracing of Atmospheric Environments*, Visual Computer, p104-113, 1991.

In our algorithm, the two phenomena are treated differently. The isotropically scattered part of the radiance is simply stored in the current voxel, whereas the directionally scattered part is propagated along a new direction, computed according to the phase function using Equation 23.

Algorithm of the First Pass

For each voxel V of the volume density object O
    Initialize the isotropic radiance : $L_V = 0$.
End-For.
For each elementary sources S
    Compute the solid angle from S containing O.
    Send stochastically N rays in the solid angle.
    For each ray R
        Initialize the radiance of R : $L_R = L_S$.
        Find the first voxel V in O crossed by R.
        While $L_R >$ THRESHOLD and R is still in O
            Attenuate the radiance $L_R = e^{-\alpha\rho\delta}\ L_R$ where δ = distance in V and ρ = density of V.
            Use the probability of interception $\omega(\delta)$ to decide if scattering occurs at the current point.
            If there is not scattering then
                Propagate R with radiance L_R in the same direction.
            else
                Store the isotropic radiance in V : $L_V = L_V + rL_R$.
                Propagate the directional radiance in a stochastic direction defined by φ : $L_R = (1-r)L_R$.
            End-If.
        End-While.
        If $L_R >$ THRESHOLD Store the radiance L_R in the last encountered voxel : $L_V = L_V + L_R$.
    End-For.
End-For.

4.3 Second Pass

During the second pass of the algorithm, visualization rays are sent from the viewer toward the different pixels of the screen. Here again, a ray progresses incrementally, but this time stepping voxel by voxel (voxel sampling). At each encountered voxel, the ray accumulates the isotropic radiance stored in it during the first pass, scaled by the current attenuation of the ray. This attenuation factor is initialized to one when the ray is started from the pixel, and scaled at each step by the volume attenuation factor using the density of the current voxel and the distance covered in it.
When the attenuation of a ray comes below a specified threshold, the ray stops its progression and its radiance becomes the color of the corresponding pixel. In the current implementation, the volume density object is not placed in a complete environment including surface objects, but is only surrounded by a huge sphere used to simulate a background with a given radiance L_∞. When a ray reachs the background, the radiance L_∞ is attenuated and added to the pixel color.

Algorithm of the Second Pass

For each pixel P
    Send a visualization ray R from the eye through the pixel.
    Initialize the radiance of P : $L_P = 0$.
    Initialize the attenuation of R : $A = 1$.
    Find the first voxel V in O crossed by R.

As usual in Monte Carlo methods [3], a weighted sampling — also called *importance sampling* — for a given distribution function f can be obtained from a uniform sampling by using the inverse function F^{-1} of the repartition function F associated with f. So, any valuation of $t = F^{-1}(u)$, where u is a uniform random variable, provides a stochastic weighted value of t.
Using such a technique with previous phase functions is either impossible or leads to an extremely complex expresssion. On the contrary, for the Schlick phase function, a weighted sampling value $t \in [-1, 1]$ can be obtained from a uniform sampling value $u \in [0, 1]$ by a very inexpensive equation :

$$t = \frac{2u + k - 1}{2ku - k + 1} \quad \text{where} \quad u \in [0, 1] \tag{23}$$

4 Description of the Method

4.1 Overview

We propose a two-pass Monte-Carlo ray-tracing algorithm for simulating interaction of light in volume density objects. Such objects are modelized as tridimensional grids with constant density per voxel. The grids used in the present paper were obtained by an extension of the technique described in [13]. It generates grids of fractal densities which can be used to create convincing clouds, for instance. The shape of the object can then be controlled by solid texturing techniques [14, 15]. Finally, a varying threshold scheme, eventually combined with a turbulence function [2], can be used to simulate temporal modifications of the object.
During the first pass, which is view-independent, rays are sent from the sources and progress incrementally through the volume density object using uniform steps. At each sample point, scattering may occur, according to the probability of interception. In that case, the phase function is used as a weighting function for a Monte-Carlo sampling process in order to modify the direction of the ray. The fraction of the energy scattered isotropically is stored in the voxel, whereas the remainder is propagated along the new direction of the ray. When the ray leaves the object, its energy is stored in the last non-empty voxel it had crossed.
During the second pass, which is view-dependent, visualization rays starting from the view point are sent towards the pixels and travel straight through the scene. At each encountered voxel of a volume density object, the ray accumulates the energy stored there during the first pass, scaled by its current attenuation. Finally, when the ray leaves the scene, its radiance becomes the color of the pixel.

4.2 First Pass

During the first pass of the algorithm, the surface of each source is divided in a set of elementary disk-shaped sources [19, 9]. Rays are then sent stochastically from each elementary source, eventually according to a goniometrical distribution of the source. A ray is propagated in the scene as long as its radiance is above a specified threshold.
In the current implementation, a ray progresses incrementally, sampled in steps of uniform length (distance sampling). Other sampling strategies can be tried to reduce possible aliasing (see [4] for a complete survey). At each step, the probability of interception is computed using the distance covered from the last scattering point[2] and the average density of the object along that distance.
When there is no interception, the ray is attenuated according to the volume attenuation factor K_V (Equation 7) and then simply propagated a step further on the same direction. When there is an interception, the surface reflection factor K_S (Equation 8) describes the scattering phenomenon. In the current implementation, we use a phase function given by Equation 22 that is a normalized sum of an isotropic scattering factor ($k = 0$) and a forward scattering factor ($k' > 0$). Such a combination is in agreement with the Mie theory which states that there exists a small proportion ($r \simeq 0.1$) of isotropic scattering (corresponding ot local internal scattering), while the remainder is directional.

[2] Light sources are considered as the first scattering point.

The main characteristic of the Henyey-Bernstein phase functions is that they provide a continuum between forward scattering ($k > 0$) isotropic scattering ($k = 0$) and backward scattering ($k < 0$).

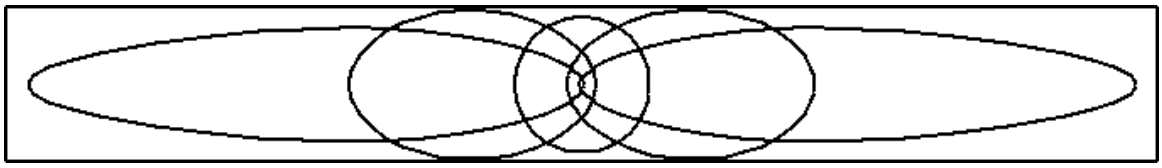

Figure 5 : Henyey-Bernstein phase function for $k = -0.9, -0.6, 0, 0.6, 0.9$

Moreover, by taking a normalized sum of several $\varphi_k(t)$ with different values for k, very general phase functions can be obtained :

$$\varphi_k(t) = \sum_{i=1}^{n} r_i \; \varphi_{k_i}(t) \quad \text{where} \quad \sum_{i=1}^{n} r_i = 1 \tag{20}$$

3.5 Schlick phase function

Using the rational fraction approximation technique detailled in [18] we propose the following phase function familly :

$$\varphi_k(t) = \frac{1 - k^2}{(1 - kt)^2} \quad \text{where} \quad k \in \,]-1, 1[\tag{21}$$

The new formulation presents several advantages compared to previous ones. First, as with the Henyey-Bernstein functions, a continuum is insured between perfectly backward scattering ($k = -1$) isotropic scattering ($k = 0$) and perfectly forward scattering ($k = 1$), but the functions are much faster to compute (1 subtraction, 2 multiplications and 1 division in an optimized implementation).

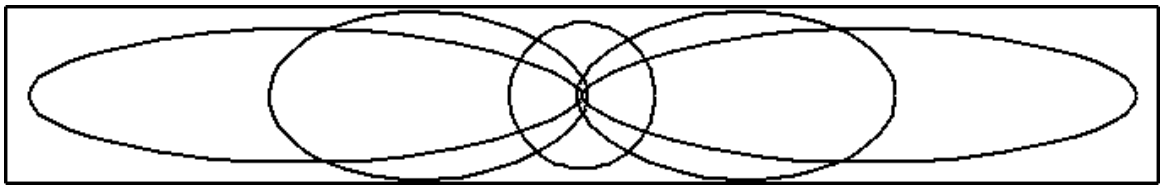

Figure 6 : Schlick phase function for $k = -0.95, -0.8, 0, 0.8, 0.95$

Second, when using a normalized sum of two independent phase functions φ_k and $\varphi_{k'}$, very close approximations of theoretical functions can be found as showed in Figures 2b,3b,4b :

$$\varphi_{r,k,k'}(t) = r \; \varphi_k(t) \; + \; (1 - r) \; \varphi_{k'}(t) \quad \text{where} \quad r \in [0, 1] \;\; k \in \,]-1, 1[\quad k' \in \,]-1, 1[\tag{22}$$

Rayleigh	$r = 0.50$	$k = -0.46$	$k' = 0.46$	Figure 2b
Hazy Mie	$r = 0.12$	$k = -0.50$	$k' = 0.70$	Figure 3b
Murky Mie	$r = 0.19$	$k = -0.65$	$k' = 0.91$	Figure 4b

Figure 7 : Parameter values for approximation of theoretical functions

The third advantage of the new function comes from the original use of the phase function made by our method. Indeed, whereas $\varphi(\theta)$ is generay used to express the ratio of incoming energy scattered in a direction given by angle θ, we rather use it to express the deviation of light at a scattering point. More precisely, the phase function is used as a weighting function for a Monte-Carlo process that generates the new direction of a ray after scattering.

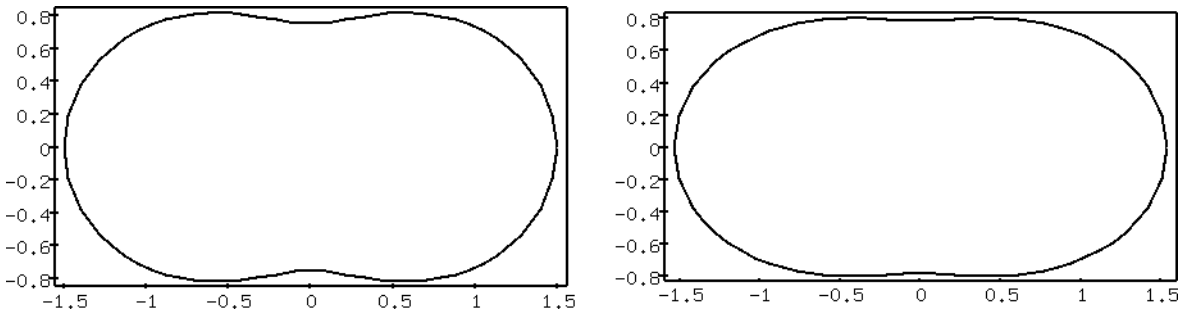

Figure 2 : Rayleigh phase function
(a) True function (b) Approximation with $r = 0.5$, $k = -0.46$ and $k' = 0.46$ (see Section 3.5)

3.3 Mie phase function

When particles are large compared to the wavelength, φ is given by the Mie theory. Two usefull approximations for foggy atmospheres are proposed in [12], the first one for hazy atmospheres and the second one for murky atmospheres.

$$\textbf{Hazy } \varphi(t) = \frac{1}{2} + \frac{9}{2}\left(\frac{1+t}{2}\right)^8 \quad \text{and} \quad \textbf{Murky } \varphi(t) = \frac{1}{2} + \frac{33}{2}\left(\frac{1+t}{2}\right)^{32} \tag{18}$$

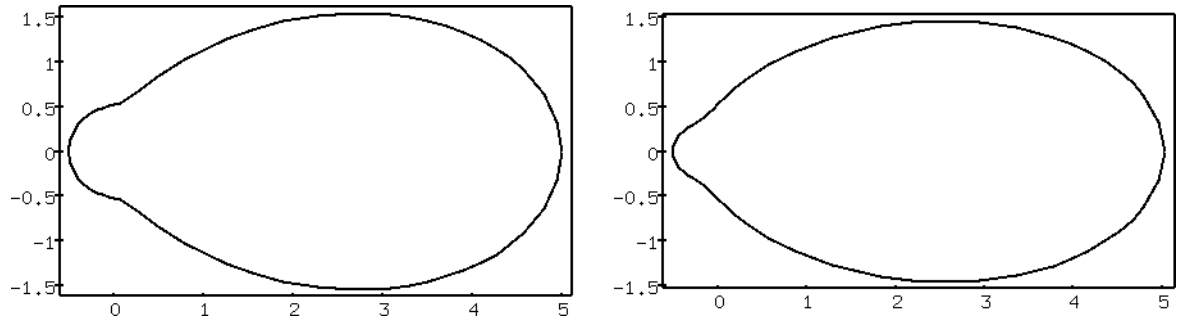

Figure 3 : Hazy Mie phase function
(a) True function (b) Approximation with $r = 0.12$, $k = -0.5$ and $k' = 0.7$ (see Section 3.5).

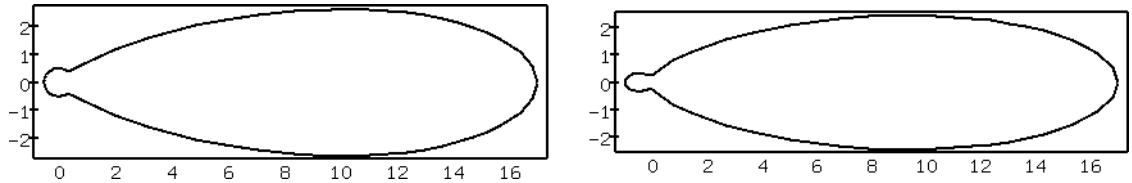

Figure 4 : Murky Mie phase function
(a) True function (b) Approximation with $r = 0.19$, $k = -0.65$, $k' = -0.91$ (see Section 3.5).

3.4 Henyey-Bernstein phase function

A familly of phase functions that has shown a good match to experimental data has been proposed by Henyey and Bernstein :

$$\varphi_k(t) = \frac{1-k^2}{(1-2kt+k^2)^{1.5}} \quad \text{where} \quad k \in]-1, 1[\tag{19}$$

In Equation 9, $\omega(P, P')$ is function of the wavelength, due to the spectral dependence of σ. For algorithmic reasons (a ray carries several spectral samples), it is more convenient to have a wavelength independent probability of interception. Therefore, we rather use an average scattering coefficient $\overline{\sigma}$ for the definition of $\omega(P, P')$:

$$\omega(P, P') = 1 \; - \; e^{-\int_P^{P'} \overline{\sigma} \; \rho(P'') \; dP''} \tag{10}$$

Finally, in the particular case where density between P and P' is constant, Equation 10 becomes :

$$\omega(\delta) = 1 \; - \; e^{-\overline{\sigma} \; \rho \; \delta} \tag{11}$$

where ρ is the density and δ the distance between P and P'. Notice that when $\rho = 0$ (outside the object, for instance) the probability of interception is null, as expected.

3 Phase Functions

The phase function $\varphi(V, V')$ expresses the ratio of energy propagated in direction V compared to the energy coming from direction V'. It satisfies two important properties that result directly from physics of light. First, due to the *Helmholtz Reciprocity Rule*, φ is symmetric relative to V and V' :

$$\forall \; V \in \mathcal{V} \quad \forall \; V' \in \mathcal{V} \quad \varphi(V, V') = \varphi(V', V) \tag{12}$$

Second, due to the *Energy Conservation Law*, φ has to fulfill the normalization condition :

$$\forall \; V \in \mathcal{V} \quad \frac{1}{4\pi} \int_{V' \in \mathcal{V}} \varphi(V, V') \; dV' = 1 \tag{13}$$

Moreover, φ is usually symmetric around the incident direction of light and so $\varphi(V, V')$ depends only on the angle θ between V and V'. Therefore Equation 13 can be written :

$$\frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} \varphi(\theta, \gamma) \; \sin\theta \; d\theta \; d\gamma = \frac{1}{2} \int_0^{\pi} \varphi(\theta) \; \sin\theta \; d\theta = 1 \tag{14}$$

Finally $t = \cos\theta$ gives the following normalization condition :

$$\int_{-1}^{1} \varphi(t) \; dt = 2 \tag{15}$$

Let us briefly recall several phase functions detailled in [1].

3.1 Isotropic phase function

The simplest expression for φ is to suppose isotropic scattering :

$$\varphi(t) = 1 \tag{16}$$

3.2 Rayleigh phase function

When particles are small compared to the wavelength, φ is given by the Rayleigh theory :

$$\varphi(t) = \frac{3}{4} \; (1 + t^2) \tag{17}$$

We consider here the most important phenomena which are involved in physics of participating media : *absorption* and *scattering* [6]. Absorption corresponds to the transformation of energy from the visible spectrum into warmth. Scattering corresponds to a distribution of energy in the whole space when a ray of light intercepts the surface of a particle — such a distribution is usually described by a so-called *phase function*. Both phenomena are responsible for the attenuation of energy in a participating medium, as a function of the distance. Bouguer's law expresses the attenuation $A(P, P')$ between two points P and P' :

$$A(P, P') = e^{-\tau(P,P')} \tag{3}$$

where $\tau(P, P')$ is the optical depth between P and P'. The optical depth can be defined by an absorption coefficient α_o and a scattering coefficient σ_o :

$$\tau(P, P') = \int_P^{P'} \alpha_o(P'')\ dP'' \quad + \quad \int_P^{P'} \sigma_o(P'')\ dP'' \tag{4}$$

In the case of volume density objects, it is convenient to use absorption and scattering coefficient depending on the material density ρ [17] :

$$\sigma = \sigma_o/\rho \quad \text{and} \quad \alpha = \alpha_o/\rho \tag{5}$$

So the optical depth can be rewritten depending on the density of the object :

$$\tau(P, P') = \int_P^{P'} (\alpha + \sigma)\ \rho(P'')\ dP'' \tag{6}$$

In our presentation, the rendering equations describe what happens between two consecutive scattering points. So the volume attenuation factor $K_V(P, P')$ expresses only the absorption that occurs between point P and P' :

$$K_V(P, P') = e^{-\int_P^{P'} \alpha\ \rho(P'')\ dP''} \tag{7}$$

The surface reflection factor $K_S(P, V, V')$ can also be naturally interpreted. It expresses the scattering at a point P where a ray intercepts the surface of a particle and can be decomposed into a wavelength-dependent term and a direction-dependent one :

$$K_S(P, V, V') = \frac{1}{4\pi}\ \frac{\sigma}{\alpha + \sigma}\ \varphi(V, V') \tag{8}$$

where $\frac{\sigma}{\alpha+\sigma}$ is the reflectivity — also called *albedo* — and $\varphi(V, V')$ the phase function of the object.
At every point P where there is no scattering, $\varphi(V, V')$ is a Dirac function equal to one when $V = V'$ and zero otherwise. At scattering points, $\varphi(V, V')$ is the true phase function for which several expressions will be given in Section 3. Notice that such an interpretation allows us to treat solid objects and volume density objects in a similar way. Indeed, for solid objects the phase function can be viewed as the bidirectional reflectance/transmittance distribution function.
Equation 8 and Equation 7 modelize the physical phenomena that occur respectively at scattering points and between two consecutive scattering points. Now we have to find the points where scattering takes place. For this, we use a *probability of interception* which increases as a function of the density, the scattering coefficient and the distance covered by the ray in the object since the last scattering point. In Bouguer's law, the scattering coefficient expresses the fraction of incoming energy which is lost by scattering. Therefore it is natural to define the probability of interception $\omega(P, P')$ as :

$$\omega(P, P') = 1\ -\ e^{-\int_P^{P'} \sigma\ \rho(P'')\ dP''} \tag{9}$$

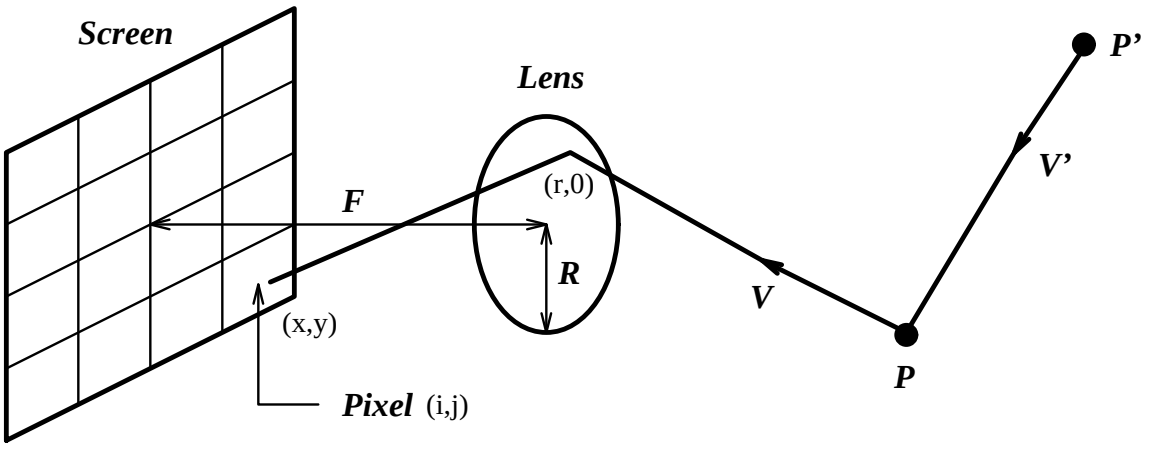

Figure 1 : Image rendering and scene rendering equations

$$I(i,j) = \frac{1}{T\pi R^2} \int_x \int_y \int_r \int_\theta \int_t K_V(M,P)\, L(P,V)\, dx\, dy\, r\, dr\, d\theta\, dt \tag{1}$$

$$L(P,V) = L_E(P,V) + \int_{V' \in \mathcal{V}} K_S(P,V,V')\, K_V(P,P')\, L(P',V')\, dV' \tag{2}$$

- $I(i,j)$: Illumination of pixel (i,j)
- $L(P,V)$: Radiance leaving point P in direction V
- $L_E(P,V)$: Self-emitting radiance leaving point P in direction V
- $K_S(P,V,V')$: Surface reflection factor at point P between directions V and V'
- $K_V(P,P')$: Volume attenuation factor between points P and P'
- $\mathcal{V}$: Set of directions for the incident light (solid angle 4π)
- $T\pi R^2$: Normalization factor (where T is the shutter time and R is the lens radius)
- dx : Differential length element along the x axis of the pixel ($i \leq dx \leq i+1$)
- dy : Differential length element along the y axis of the pixel ($jleqdy \leq j+1$)
- dr : Differential radius element on the lens ($0 \leq dr \leq R$)
- $d\theta$: Differential angle element on the lens ($0 \leq d\theta \leq 2\pi$)
- dt : Differential time element ($0 \leq dt \leq T$)
- dV' : Differential solid angle element surrounding direction V'

These equations are presented here using the notations proposed in [18] and are generalized to account for participating media. The equations are monochromatic so they should be evaluated, theoretically for each wavelength of the visible spectrum, and practically for a limited number of spectrum samples [11]. Moreover, the equations use geometrical optics assumptions and therefore do not consider phenomena coming from physical optics (diffraction, interference, polarization).
The $K_S(P,V,V')$ coefficient — called *surface reflection factor* — is the ratio of radiances between directions V and V' at a scattering point P. The $K_V(P,P')$ coefficient — called *volume attenuation factor* — is the ratio of radiances between two scattering points P and P'. In the image (resp. scene) rendering equation, the point P (resp. P') is the last scattering point before point M (resp. P).

A Rendering Algorithm for Discrete Volume Density Objects

Philippe Blasi Bertrand Le Saëc Christophe Schlick

LaBRI [1]
351, Cours de la Libération
33405 Talence (FRANCE)
[lesaec,schlick]@labri.greco-prog.fr

Abstract : We present a new algorithm for simulating the effect of light travelling through volume objects. Such objects (haze, fog, clouds...) are usually modelized by voxel grids which define their density distribution in a discrete tridimensional space. The method we propose is a two-pass Monte-Carlo ray-tracing algorithm that does not make any restrictive assumptions neither about the characteristics of the objects (both arbitrary density distributions and phase functions are allowed) nor about the physical phenomena included in the rendering process (multiple scattering is accounted for). The driving idea of the algorithm is to use the phase function for Monte-Carlo sampling, in order to modify the direction of the ray during scattering.

Keywords : Monte-Carlo Ray-Tracing, Participating Media, Volume Density Objects, Multiple Scattering.

1 Introduction

Rendering methods of objects defined by their density distribution in a tridimensional space can be divided in view-dependent and view-independent solutions. Generally, these algorithms make some restrictive assumptions either on the characteristics of the objects or on the physical phenomena included in the rendering process. For instance, view-dependent techniques (which use either polygonal projective or ray-tracing approaches) [1, 10, 12, 17, 5] consider only single scattering and sometimes only specific (*eg* constant or linear) density distributions. On the other hand, the unique view-independent technique (which uses a radiosity approach) [16] accounts for multiple scattering but is limited to isotropic phase functions. In [8], a very complete ray-tracing algorithm is presented including a possible extension to multiple scattering. Both [8] and [16] are very close to the physical behaviour of light in a volume density object, and therefore can achieve very realistic pictures, but unfortunately these methods require large computing times.
In the present paper, we propose an inexpensive two-pass Monte-Carlo ray-tracing algorithm that enables rendering of volume objects with arbitrary density distributions, arbitrary phase functions and arbitrary levels of scattering. Section 2 provides definitions and notations, Section 3 details usefull phase functions, Section 4 describes the method and Section 5 concludes with experimental results.

2 Rendering Equations

The process of generating realistic pictures consists in the resolution of two equations : the *image rendering equation* (Equation 1) and the *scene rendering equation* (Equation 2). The first one defines the illumination of a given pixel in the image. It is essentially a mean over several integration dimensions (pixel width, pixel height, lens radius, lens angle, shutter time). The second defines the radiance of each couple (P, V) where P is a point and V a direction of the scene. It is a Fredholm equation that expresses the light transport in the environment.

[1] Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux I* and *Centre National de la Recherche Scientifique*) . The present work is also granted by the *Conseil Régional d'Aquitaine*.