A Survey on Participating Media Rendering Techniques

Eva Cerezo, Frederic Perez-Cazorla, Xavier Pueyo, Francisco Seron, François X. Sillion

The Visual Computer manuscript No.
(will be inserted by the editor)

A Survey on Participating Media Rendering Techniques

Eva Cerezo[1], **Frederic Pérez**[2], **Xavier Pueyo**[2], **Francisco J. Seron**[1], **François X. Sillion**[3]

[1] Advanced Computer Graphics Group (GIGA), Computer Science Department, University of Zaragoza, Spain. e-mail: ecerezo, seron@unizar.es.
[2] Girona Graphics Group (GGG), Computer Science Department, University of Girona, Spain. e-mail: fredericpcx@terra.es, xavier.pueyo@ima.udg.es
[3] ARTIS- GRAVIR Laboratory, INRIA, France. e-mail: Francois.Sillion@imag.fr

The date of receipt and acceptance will be inserted by the editor

Abstract Rendering participating media is important for a number of domains, ranging from commercial applications (entertainment, virtual reality) to simulation systems (driving, flying and space simulators) and safety analyses (driving conditions, sign visibility). This article surveys global illumination algorithms for environments including participating media. It reviews both appearance-based and physically-based media methods, including the single scattering and the more general multiple-scattering techniques. The objective of the survey is the characterization of all these methods: Identification of their base techniques, assumptions, limitations and range of utilization. We conclude with some reflections about the suitability of the methods depending on the specific application involved, and possible future research lines.

Key words Three-dimensional graphics and realism, shading, radiosity, raytracing, participating medium, global illumination.

1 Introduction

This paper surveys rendering algorithms for environments including participating media. This survey is an extended and updated version of a previous one done by the authors [70] (that focused only on multi-scattering methods) and of a first one written by Max [56]. The generation of physically accurate images of participating media is an extremely challenging computational problem. Two types of difficulties arise when treating light propagation in this type of environments. First of all, interaction phenomena take place not only in the medium boundaries but within any point of the medium. Therefore, optical properties in each point of the medium have to be known, and not only radiances on surfaces but source radiances throughout the space have to be computed. A second difficulty comes from the spectral dependence of the medium characteristic parameters so that a detailed spectral analysis is usually needed. Nevertheless, phenomena like fluorescence or phosphorescence, which imply a transfer of energy from one wavelength to another are not significant in the wavelengths corresponding to the visible range of the light spectrum. Therefore, we do not consider them in this survey. Additionally, we restrict ourselves to still images and static environments, i.e. we deal neither with walkthroughs nor animated scenes.

In this section we first present the transport equation governing the transfer of energy in participating media. Then we briefly present the most widely used medium models and discuss applications related to the rendering of participating media. The rest of the survey is structured as follows: Section 2 presents appearance- based methods (that we call *fake media methods*). Section 3 focuses on single scattering methods and Section 4 on methods that deal with the more complex problem of multiple scattering. Finally we propose some reflections on the choice of participating media methods and future research lines.

1.1 Participating Media: The Transport Equation

As radiation travels through a participating medium it undergoes three kinds of phenomena: absorption, scattering and emission. (see Fig. 1) [15, 86].

Absorption consists of the transformation of radiant energy into other energy forms. For a differential distance dx, the relative reduction of radiance is given by $\kappa_a(x)\,dx$, $\kappa_a(x)$ being the *coefficient of absorption* of the medium at point x. Scattering means a change in the radiant propagation direction. It is generally divided into out-scattering and in-scattering. Out-scattering reduces the radiance in the particular direction along dx by the factor $\kappa_s(x)\,dx$, $\kappa_s(x)$ being the *scattering coefficient*. Mathematically the reduction of radiance is expressed as $dL(x) = -\kappa_t(x)\,L(x)\,dx$, where $\kappa_t = \kappa_a + \kappa_s$ is the *extinction coefficient*. The solution of the previous differential equation is *Bouguer's law* (also known as the *Beer's law*):

$$L(x) = L(x_0)\underbrace{e^{-\int_{x_0}^{x}\kappa_t(u)\,du}}_{\tau(x_0,x)} = L(x_0)\,\tau(x_0,x)\,, \qquad (1)$$

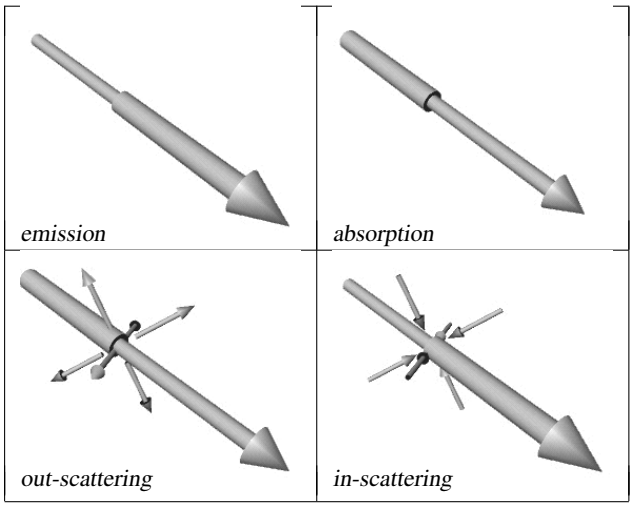

Fig. 1 Interaction of light in a participating medium.

$\int_{x_0}^{x} \kappa_t(u)\,du$ is called the *optical thickness*, and $\tau(x_0,x)$ the *transmittance* from x_0 to x. Notice that Bouguer's law simply models the reduction of radiance due to out-scattering and absorption. Notice also that, for simplicity, we are omitting the frequency dependency from the expressions. Table 1 summarises relevant terms used in this survey.

Table 1 Table of terms.

Symbol	*Meaning*
$L(x,\omega)$, $J(x,\omega)$	Radiance and source radiance ($\mathrm{W\,m^{-2}\,sr^{-1}}$)
$p(\omega_o,\omega_i)$	Phase function
x	A point in $\mathbb{R}^3$
$\kappa_a(x)$	Absorption coefficient ($\mathrm{m^{-1}}$)
$\kappa_s(x)$	Scattering coefficient ($\mathrm{m^{-1}}$)
$\kappa_t(x)$	Extinction coefficient ($\mathrm{m^{-1}}$)
$\rho_{bd}(x,\omega_o,\omega_i)$	Bidirectional Reflectance Distribution Function (BRDF)
$\tau(x,y)$, $\tau_a(x,y)$	Transmittance, trans. due to absorption
ω	Direction
$d\sigma_\omega$	Differential solid angle (sr)
$\Omega(x)$	Scattering albedo
S, H	Sphere, hemisphere of directions

In order to derive the transport equation, radiance increases due to emission and in-scattering have to be taken into account. Emission refers to the process of creation of radiant energy. Radiance along the propagation direction is also augmented because of in-scattering, i.e. because of light impinging on x that is scattered into the considered direction. The spatial distribution of the scattered light is modeled by the *phase function* $p(\omega_o,\omega_i)$. The phase function has the physical interpretation of being the scattered intensity in direction ω_o, divided by the intensity that would be scattered in that direction if the scattering were isotropic (i.e. independent of the direction). Phase functions in Computer Graphics are usually symmetric around the incident direction, so they can be parameterized by the angle between the incoming and outgoing direction. Different phase functions have been proposed to model different media. The simplest phase function is the isotropic one (constant) and represents the counterpart of the diffuse BRDF for participating media; this is why it is used in the zonal method (section 4.1.1). Rayleigh phase functions are used to model scattering processes produced by spherical particles whose radii are smaller than about one-tenth the radiation wavelength. This is the case of the particles that constitute the smoke of cigarettes and of the gas molecules of the atmosphere. On the other hand, Mie phase functions are generally used for scattering where the size of the particles is comparable to the wavelength of light. It is applied to many meteorological optics phenomena like the scattering by particles responsible for the polluted sky, haze and clouds. Mie phase functions are generally complex and heavily depend on the particles' size and conductivity. There are several approximations to Mie phase functions, such as the hazy atmosphere approximation, for sparse particle densities, or the murky atmosphere approximation, for dense particle ones [27]. The Henyey-Greenstein (HG) phase functions are another mathematically simple approximation to Mie phase functions, with a parameter determining the eccentricity of their elliptical shape. This functions have been extensively used in the spherical harmonics and discrete ordinates methods (sections 4.1.3 and 4.1.2). Finally, the Schlick phase functions are similar to the HG ones but faster to compute and very well suited to be used in Monte Carlo methods (section 4.2). In Fig. 2 several Schlick approximations are shown.

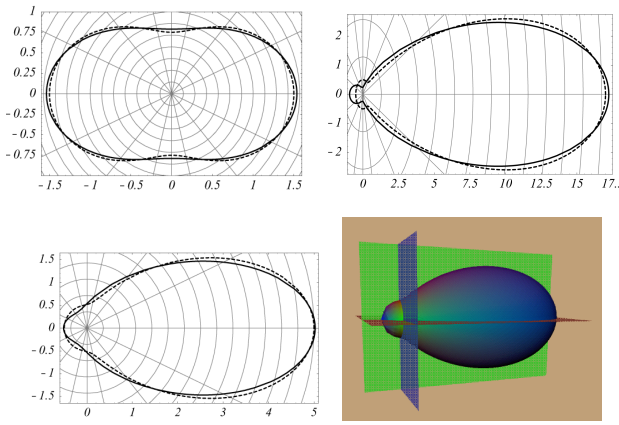

Fig. 2 Examples of Schlick [9] approximations to some phase functions.Top Left: Schlick approximation (solid line) of the Rayleigh phase function. Top Right: Schlick approximation (solid line) of Murky atmosphere approximation (dashed line). Bottom Right: Schlick approximation (solid line) of Hazy atmosphere approximation (dashed line).Bottom Left: 3D representation of the latter Schlick approximation.

The *transport equation* takes all these phenomena into account and describes the variation of radiance in dx around x in direction ω_o (note that the ω_o parameter is omitted from the radiances in the following expressions to improve readability):

$$\frac{dL(x)}{dx} = \kappa_t(x)\,J(x) - \kappa_t(x)\,L(x) \tag{2}$$
$$= \underbrace{\kappa_a(x)\,L_e(x)}_{\text{emission}} + \underbrace{\frac{\kappa_s(x)}{4\pi}\int_S L(x,\omega_i)\,p(\omega_o,\omega_i)\,d\sigma_{\omega_i}}_{\text{in-scattering}}$$
$$- \underbrace{\kappa_a(x)\,L(x)}_{\text{absorption}} - \underbrace{\kappa_s(x)\,L(x)}_{\text{out-scattering}},$$

where S denotes the set of directions on the sphere around point x and $J(x)$ is the *source radiance*, which describes the local production of radiance, i.e. the radiance added to the point x due to self-emission and in-scattering. Concretely,

$$J(x) = \underbrace{(1-\Omega(x))\,L_e(x)}_{J_e(x)} + \frac{\Omega(x)}{4\pi}\int_S L(x,\omega_i)\,p(\omega_o,\omega_i)\,d\sigma_{\omega_i}, \tag{3}$$

where $\Omega = \frac{\kappa_s}{\kappa_t}$ is the so-called *scattering albedo*—sometimes also called *single scattering albedo*.

The integrated form of Equation 2, called the *integral transport equation* (Fig. 3), is:

$$L(x) = \underbrace{\tau(x_0,x)\,L(x_0)}_{L_{ri}(x)} + \underbrace{\int_{x_0}^{x}\tau(u,x)\,\kappa_t(u)\,J(u)\,du}_{L_m(x)}, \tag{4}$$

$L_{ri}(x)$ being the *reduced incident radiance*, due to the radiance of a background surface (if any), and $L_m(x)$ the *medium radiance*, due to the contribution of the source radiance within the medium [93]. By means of the change of variable $t = \int_{x_0}^{x}\kappa_t(u)\,du$, Equation 4 can be rewritten more compactly [80]:

$$L(t) = e^{-t}\left[L_0 + \int_0^t J(v)\,e^{v}\,dv\right].$$

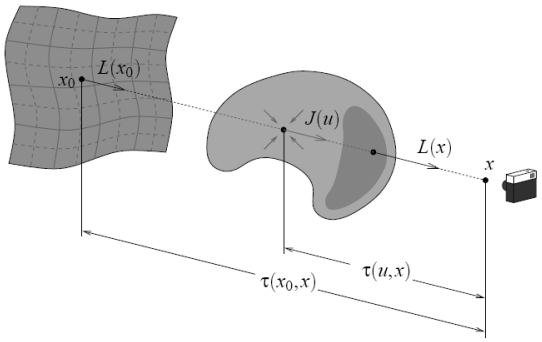

Fig. 3 The integral transport equation: the radiance $L(x)$ at point x in a given direction is the sum of the reduced incident radiance $\tau(x_0,x)\,L(x_0)$ and the contribution of the source radiance within the medium.

The source radiance can be decomposed into three terms, accounting for self-emission, for the (first) scattering of reduced incident radiance, and for the scattering of the medium radiance. Mathematically this is expressed as follows:

$$J(x) = J_e(x) + \underbrace{\frac{\Omega(x)}{4\pi}\int_S L_{ri}(x,\omega_i)\,p(\omega_o,\omega_i)\,d\sigma_{\omega_i}}_{J_{ri}(x)}$$
$$+ \underbrace{\frac{\Omega(x)}{4\pi}\int_S L_m(x,\omega_i)\,p(\omega_o,\omega_i)\,d\sigma_{\omega_i}}_{J_m(x)}. \tag{5}$$

The boundary conditions of the integral transport equation (Equation 4) are represented by the *global illumination equation* [89]:

$$\underbrace{L(x,\omega_o)}_{\text{total radiance}} = \underbrace{L_e(x,\omega_o)}_{\text{emitted radiance}} + \underbrace{\int_H \rho_{bd}(x,\omega_o,\omega_i)\,L_i(x,\omega_i)\cos\theta_i\,d\sigma_{\omega_i}}_{\text{reflected radiance}}, \tag{6}$$

where $L(x,\omega_o)$ is the radiance leaving point x in direction ω_o, $L_e(x,\omega_o)$ is the self-emitted radiance, $L_i(x,\omega_i)$ is the incident radiance from direction ω_i, $\rho_{bd}(x,\omega_o,\omega_i)$ is the bidirectional reflectance distribution function (BRDF) modeling the reflective properties at x, H is the set of directions of the hemisphere above x, θ is the angle between the normal of the surface at x and ω_o, and $d\sigma_{\omega_i}$ is the differential solid angle corresponding to direction ω_i. The equation states that a set of radiating surfaces reach energy equilibrium when the sum of the reflected and the transmitted (or absorbed) energy equals the incident energy.

The goal of rendering algorithms is the resolution of the integral transport equation (Equation 4) and the global illumination equation (Equation 6), at least for the points and directions visible by the camera.

Before beginning the review of the solution methods, two simplified situations are discussed: The *no scattering* and the *single scattering* cases.

1.1.1 The No Scattering Case This particular case is given when the participating media do not scatter (i.e. when $\kappa_s = 0$). For example, in the first steps of an explosion there is a high emission of light and all kinds of scattering effects can be neglected [99]. Another example is the rendering of fire. Fire is a blackbody radiator (it absorbs but does not scatter) that creates low albedo smoke [58]. Under this assumption, $\kappa_t = \kappa_a$, $\Omega(x) = 0$ and $J(x) = L_e$, and the integral transport equation (Equation 4) is considerably simplified:

$$L(x) = \tau_a(x_0,x)\,L(x_0) + \int_{x_0}^{x}\tau_a(u,x)\,\kappa_a(u)\,L_e(u)\,du, \tag{7}$$

with the transmittance term $\tau_a(x_0,x)$ being $\exp(-\int_{x_0}^{x}\kappa_a(u)\,du)$.

Under the non-scattering assumption, a simple depth of field effect can be achieved through the use of a homogeneous non-emitting medium—κ_a being constant and $L_e = 0$. In this case Equation 7 reduces to

$$L(x) = e^{-\kappa_a\|x_0-x\|}\,L(x_0).$$

A more interesting effect can be obtained if the L_e term of Equation 7 is supposed to be the scattering of a constant ambient illumination in a homogeneous non-emitting medium. This would be a rough approximation of the single scattering case (see Equations 9 and 10):

$$L(x) = e^{-\kappa_a \|x_0 - x\|} L(x_0) + (1 - e^{-\kappa_a \|x_0 - x\|}) L_e \,. \qquad (8)$$

1.1.2 The Single Scattering Case When the participating medium is optically thin (i.e. the transmittance through the entire medium is nearly one) or has low albedo, then the source radiance can be simplified to ignore *multiple scattering* within the medium, considering this term—$J_m(x)$ in Equation 5—negligible. Representations for both the single and the multiple scattering cases are depicted in Fig. 4. Under the single scattering assumption, at point x, the contribution of the scattering of the medium radiance $J_m(x)$ to the source radiance is set to zero, considering that x is the first scattering point of the radiance coming from the background surfaces. The expressions for the source radiance (Equations 3 and 5) and the integral transport equation (Equation 4) in this case are simplified as follows:

$$J(x) \approx J_{ss}(x) = J_e(x) + \frac{\Omega(x)}{4\pi} \int_S L_{ri}(x, \omega_i)\, p(\omega_o, \omega_i)\, d\sigma_{\omega_i} \,, \qquad (9)$$

$$L(x) = \tau(x_0, x) L(x_0) + \int_{x_0}^{x} \tau(u, x)\, \kappa_t(u)\, J_{ss}(u)\, du \,. \qquad (10)$$

Single scattering represents a high simplification with respect to multiple scattering. Setting the general conditions under which the single scattering criterion is satisfied is difficult; it is not valid, for example, for clouds because of their high albedo [13], being in the [0.7–0.9] range for cumulus and stratus [60]; also Blinn states that multiple scattering cannot be neglected when $\Omega > 0.3$ [11]. Van de Hulst recognizes that even the simplest law of scattering for the individual particles (isotropic scattering) leads to complex mathematics in the multiple scattering problem [32].

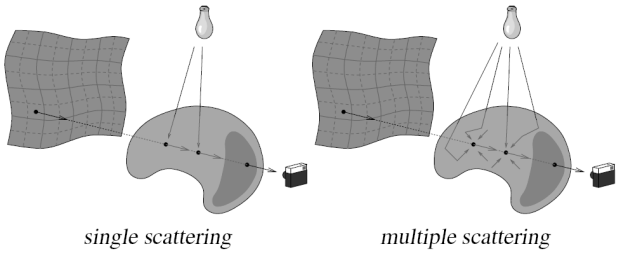

Fig. 4 Schematic representations for the single and multiple scattering cases.

1.2 Participating Media Models

A model for a participating medium must allow the definition of medium emittance, phase function, extinction coefficient and scattering albedo as functions of position in the medium. It is also possible to use other combinations of media coefficients from which the extinction coefficient and the scattering albedo can be derived. This includes the utilization of mass coefficients instead of the linear coefficients (absorption, scattering and extinction) we introduced in Section 1.1. Mass coefficients are simply the result of dividing the corresponding linear coefficient by the material density [86].

Unlike the case of surfaces, whose geometric and reflectance properties can be treated separately, the definition of the geometry and the optical properties for participating media are tightly related. If the extinction coefficient or the densities of the particles of the medium are given directly as a function of position in space, then the geometry of the medium is implied. Different representations have been used for participating media. The simplest case is an homogeneous all pervading volume, or layers with constant properties—usually termed as *constant density medium* [11, 54, 44, 62, 106]. For inhomogeneous media heuristic functions have been used [25, 71, 22], as well as texturing functions and fractal algorithms [25, 22, 81, 82], particle systems [107], and blobs [92, 93, 61].

1.3 Applications

The rendering of images containing participating media is important for a number of applications [79]. Simulations of interest can be made for the following areas:

- Safety analyses: Smoke filled rooms (visibility of exit signs); foggy environments (roadway lighting, relative contrast of target objects such as traffic signs in foggy driving).
- Military: Remote sensing (atmospheric effects attenuate and blur images of land surfaces acquired by distant sensors); underwater vision; battlefield smoke plumes.
- Industrial: Design of efficient headlamps for foggy driving.
- Commercial: Entertainment, virtual reality.
- Visual simulation systems for the training of drivers of cars or ships for which optical effects in participating media are of particular importance; also fire fighter training.

The need for making use of parameters like those characteristic of the rendering of participating media (the absorption coefficient, the albedo or the phase function) appears in other Computer Graphics domains. An example could be the construction of approximate BRDFs of layered materials like paint coating and skin (see for example [28, 40] for subsurface scattering and [39] for rendering wet materials).

2 Fake Media Methods

Some participating media rendering methods do not take into account the physical phenomena involved—although they ob-

tain realistic (visually pleasant) results. This is completely acceptable for certain applications, such as the ones mentioned below. We term those methods that are not physically based but rather appearance based *fake media methods*. These include the work by Gardner [25], who developed a model to generate synthetic clouds for economical visual simulation with enough realism for a wide variety of complex cloud formations. Gardner uses a 2D model with a *sky plane* far enough from the viewer, and a 3D model with ellipsoids and a mathematical function to generate textures. Perlin [71] focuses on naturalistic visual complexity in a wider range, introducing a *Pixel Stream Editing language* to create representations of clouds, fire, water, marble, etc., by means of noise and turbulence functions. Yaeger et al. [107] have developed a method to display a simulation of the atmospheric flux of the atmosphere of Jupiter for the film 2010. This method mixes physical simulation of fluids dynamics (for the movement of flow field of the atmosphere) and visual simulation (a two-dimensional particle model to generate textures that are mapped onto a polygonalized sphere). Since the observer is restricted to be far away from the planet, its atmosphere is considered opaque, and thus there is no participating media treatment. In fact, Blinn [11] states that the cloudy surface of Jupiter follows the ideal Lambert law very closely.

Premoze and Ashikmin [73] have obtained impressive images of natural waters (Fig. 5). Even though their model considers light transport in water they have focused on their work on the appearance of the water surface so that important simplifications are made. A one-dimensional equation is solved: radiance in the medium depends only on the depth (z). They do not try to simulate the radiance due to scattering: a diffuse field radiance L_{df} that comprises the combined effect of light scattering throughout the media (taken as a uniform water body) is considered. They use an equation similar to Equation 8: $L(0) = e^{-\kappa_t R} L(z) + (1 - e^{(-\kappa_t + \kappa_d \cos\theta)R}) L_{\mathrm{df}}(0)$ where $L(0)$ is the apparent radiance just below the air-water interface, $L(z)$ is the radiance of the target at depth z, $R = -z/\cos\theta$ is the total path length (angle θ counted from vertical direction (positive z-axis) and κ_d is an apparent water diffuse attenuation coefficient. $L_{\mathrm{df}}(0)$ is the diffused radiance just below the sea surface and is estimated using empirical equations that relate it with downwelling irradiance due to sun and skylight.

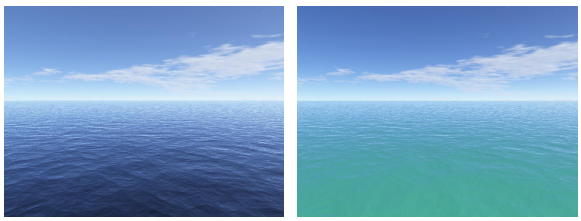

Fig. 5 Rendering different types of water: deep ocean (left) and tropical water (right) by Premoze et Ashikhmin [73].

In the context of real-time animation of fog, Biri et al. [8] have introduced a model of fog where a set of well-chosen functions for the extinction coefficient κ_t—functions that allow analytical integration—are used to achieve a realistic fog. Considering a constant source radiance within the fog (J_{fog}), Biri et al. extend Equation 8 for non-uniform fog: $L(x) = \tau(x_0,x)\,L(x_0) + (1 - \tau(x_0,x))\,J_{\mathrm{fog}}$. The transmittances $\tau(x_0,x)$ are efficiently computed because of the definition of κ_t, and the rendering schema uses graphics hardware to achieve animation in real-time.

The methods presented in this section are not suited to a proper consideration of self-shadowing or shadows cast by the medium due to different reasons. Those methods based on textures [25, 71, 107], focus on the modeling of the medium but not on its interaction with the scene. Other methods do not need to consider them due to the specific problem they are trying to solve: this is the case of the sea surface images of Premoze and Ashikmin [73] or the fog scenes generated with Biri's method [8], that just uses color blending without integrating the fog in a global illumination scheme.

3 Single Scattering

Since the early 80's, many single scattering methods for realistic rendering have been proposed, focusing on a set of different phenomena including participating media. The single scattering category groups together a variety of methods, including the early historical ones [11, 41, 54, 62, 44] but also more recent methods that focus on obtaining fast and pleasant, yet physically-based, participating media images. We classify the resolution methods in three categories: analytic (an analytic solution is possible due to the strong simplifications made in the model), deterministic (based on numerical solutions) and stochastic (which introduces some kind of random sampling). We point out two important features regarding these methods. First, they are based on the single-scattering case (Section 1.1.2), so that they all impose very strong limitations on the media, limitations that are, in general, not realistic. Secondly, they usually propose solutions that are strongly related to the specific medium or problem they are trying to solve, so that it is difficult to apply or adapt them to more general situations. This is specially true for analytic and deterministic methods; more general cases are considered in the stochastic methods. Therefore, in Table 2 we present a classification of the most representative works based on the type of resolution method and the kind of medium considered.

In what follows we review the most important works in the analytic, deterministic and stochastic categories. Subsequently, a section devoted to volume rendering has been added. Although scientific visualization is out of the scope of this work, we have included it for completeness, as it is closely related to the rendering of participating media in the case of single scattering. The section concludes with a general discussion.

Table 2 Single scattering: resolution methods and type of medium considered.

Type of Media	*Methods*		
	Analytic	Deterministic	Stochastic
General Medium		[41, 78, 22, 81]	[78, 81]
Smoke		[62, 22, 81, 95]	[81, 92]
Water		[103, 63]	
Atmosphere		[44, 33, 42, 98, 64, 34, 19]	
Fog	[106]	[54, 44, 62, 51]	
Clouds		[41, 54, 22, 81]	[81]
Light shafts		[54, 62, 19, 51]	
Other atmosph. eff.	[11]		

3.1 Analytic Methods

Blinn [11] was the first researcher concerned with the visualization of participating media, in particular for the case of cloud layers (rings of Saturn and planet atmospheres) of uniformly distributed spherical reflecting particles (constant density). Blinn solves analytically the integral transport equation (Equation 10) for this case, for single scattering, and for a light source and viewer considered to be at infinity.

Willis [106] focuses on the problem of flight simulators for scenes containing haze, mist and fog in daylight. His model is extremely simple since it does not consider light sources, but it assumes that each point in the participating media emits isotropically a certain constant power, considering isotropic scattering.

3.2 Deterministic Methods

Kajiya and Von Herzen [41], extend the Blinn single scattering model for ray tracing, eliminating all viewing and lighting restrictions—thus, the derivation of an analytic solution is infeasible. Representing the participating medium by a voxel model, the rendering procedure is separated into two steps: (1) computation of the radiance arriving at each voxel from all light sources (in other words, computation of the term L_{ri} per voxel in Equation 9), and (2) solving the eye radiance, using the intermediate results of the previous step, by an illumination model that is a discretized version of Equation 10. This illumination model was later used by Ebert and Parent [22] within a scan-line context.

Max [54] presents a scan-line based method that deals with shadow volumes to take into account the glow in haze. The radiance reaching the eye is computed by adding the contributions of the single scattering of light in the illumination volumes. Two scenarios are considered: (1) a light source at infinity illuminating a fog layer with constant or layered density with any phase function; and (2) a point light source and medium of constant density with isotropic scattering.

Nishita et al. [62] take up again Max's ideas: use of shadow and illumination volumes (for scenes composed of polygons) and integration at lit segments (Fig. 6). However, this new method is based on ray casting, while Max's one was based upon scan-line; also this is extended to deal with spotlights with angular distributions apart from parallel light sources. Boundaries between layers of constant media are defined. The scattered light contribution is computed using sampling points along the segments that traverse illumination volumes (integration segments), to diminish aliasing problems and speed-up the whole process.

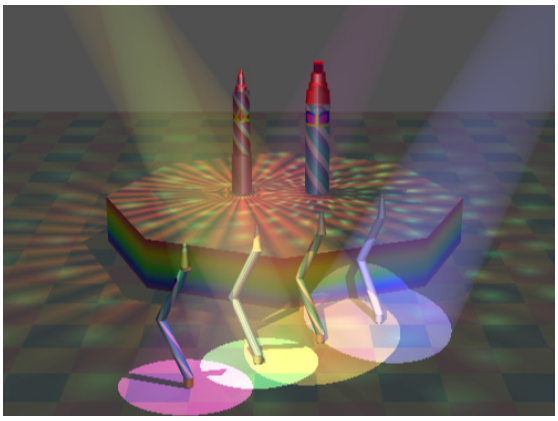

Fig. 6 Studio lighting: shafts of light in dry ice (from the animation "Feast of Light" by Nishita et al. [62]).

The model of Nishita [62] has been applied to other participating media. Stam and Fiume [95] deal with turbulent fields (i.e. wind) in gaseous phenomena and use a front-to-back strategy (a blob renderer) for rendering. Watt [103] applies Nishita's method to consider light/water interaction in his two-pass backward beam tracing method. His objective is to display underwater shafts of light [54]. Nishita and Nakamae in [63] also consider the display of different optical effects within water such as caustics, shafts of light and color of water (Fig. 7). The water is taken as a uniform particle density medium, so that numerical integration of the densities on the ray is unnecessary. Intensity due to scattered light can be obtained as an analytical function of distance between the particles and the viewpoint. In this work no ray-tracing is performed but a scanline Z-buffer and accumulation buffer are used instead. More recently, in the context of real-time rendering of participating media, Lecocq et al. [51] develop a method able to deal with mobile point light sources within a homogenous medium. Reformulating the transport equation of the model of Nishita et al. in angular terms, Lecocq et al. identify a part of the equation suitable to be approximated

by a polynomial. In order to deal with measured directional light sources, a numeric precomputation of the integrals of the developed expression is performed. For fast rendering, a 3D texture is computed (evaluating radiances sparsely at grid points) and added to the image after displaying the surfaces. The reported speed-up with respect to the method by Nishita et al. is 200 and 80 for isotropic and directional light sources respectively. The same authors have applied the method to a driving simulator that allows the testing of headlights (modeled as point light sources) in fog [52].

Another relevant work is the one of Klassen [44] which focuses on atmospheric problems and introduces spectral treatment: each pixel color is sampled at 33 wavelengths, taking samples each 10nm, to care for the selective scattering that would be wrong using a three-coordinate color space. He deals with two problems: sky color and fog. To render the sky and the sun, Klassen sets an observer on the earth ground and considers spherical layers of haze-free and haze-filled air (for molecular and particle scattering respectively). Due to the relative distances of the atmosphere and the sun, parallel light rays are assumed. Single scattering is supposed to take place only in the haze-filled layer—however, multiple scattering is not negligible for the computation of sky color, as stated by Bohren [12]. Since fog is not wavelength selective an RGB space can be used. The fog is restricted to be lit by the sun, without shadows, using parallel light rays and approximating the geometry by a flat earth.

Klassen's planar atmosphere model [44], which results in a large error near the horizon, is extended by Kaneda et al. [42], applying it to concentric spherical layers. Air molecules and aerosols are accounted for—thus Rayleigh and Mie scattering are considered—with density distributions varying exponentially with altitude. The model is used for the generation of outdoor scenes including buildings, for which sunlight and skylight are calculated. Tadamura et al. [98] continue with the rendering of outdoor scenes and propose a new method to compute the illumination at an object's location. In their method the contributions of sky light intensities are calculated at grid elements of a movable parallelepiped and the determination of whether or not each grid element is obscured by obstacles is performed using graphics hardware. The illumination at the point considered is calculated by adding the contributions due to sky elements that are not obscured by objects.

Fig. 7 Optical Effects within Water by Nishita and Nakamae [63].

Inakage [33] also considers the visualization of atmospheric effects, but tries to create a more general illumination model that could be applied to different situations (blue skies, sunsets, rainbows, shafts of light). His work is based in the so-called A-cube, in modelling the atmosphere by particles of different sizes and in the use of volume sampling techniques. The A-cube or atmospheric cube limits the modeling space to a finite volume. The density and phase functions of the scattering particles inside the cube can vary from point to point. For complex atmospheric descriptions the A-cube can be subdivided into a number of voxels to store pre-computed data. His atmospheric model handles absoption, scattering (Rayleigh and Mie) and geometric optics (refractive dispersion) but operating in then RGB colour model.

Nishita et al. [64] present a method to visualize the Earth from outer space (not from the Earth's surface towards the sky, as for example in [44, 98]), accounting for atmospheric particles (air molecules and aerosols) and the water of the sea. Their method computes the light reaching the viewpoint taking into account: (1) the light coming from the earth surface —Earth illuminated by direct sunlight and skylight—, which is affected by atmospheric scattering and (2) light scattered in the atmosphere; in the case of light coming from the surface of the sea (3) light transmission through water molecules is considered. Lookup tables are used to solve numerical integrations in (1) and (2); for (3) an analytic solution is given. The computation of the optical thickness is done by means of trapezoidal integration of sampled density—the atmosphere is non-uniformly subdivided into imaginary concentric spherical layers, established so that the density variation between two consecutive layers is below a certain threshold; using linear interpolation for the density of each sampling point. Fake clouds—that do not cast shadows onto the Earth—are added by mapping 2D fractal textures onto spherical layers. For the color of sea the light reflected on its surface and the contribution for single scattering of water molecules before being refracted towards the viewer are accounted for. The contribution of the color of the bottom of the sea is neglected, due to its depth.

Irwin [34] is also concerned with the visualization of the Earth's atmosphere. His method is similar to Nishita et al. method [64] but differs in the use of spectral wavelength sampling and reconstruction techniques (instead of working in the RGB colour model). He simulates the light of the sky and the Sun as seen from the ground or from outer space, for an atmosphere consisting of pure air, with mass density falling exponentially. Only single scattering and Rayleigh scattering are considered—neglecting the light from the sky (not the sun) reflected from the Earth's surface and ozone absorption. With these restrictions, an atmosphere modeled as a spherical shell, the question is to find the appropriate line segments corresponding to the geometry to compute illumination paths, paths that are solved by numerical integration.

More recently, Dobashi et al. [19] have applied a method again similar to the one of Nishita et al. [64] but aiming at ob-

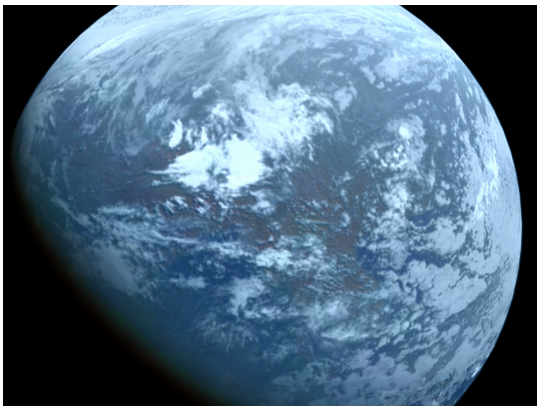

Fig. 8 Rendering earth's atmosphere using graphics hardware, by Dobashi et al. [19]

taining appropriate textures (Fig. 8). The final goal is to make the interactive rendering of atmospheric effects possible, by means of graphics hardware. They also present a method that approximates multiple scattering by an ambient term to obtain images of typical shafts of light caused by sunlight.

3.3 Stochastic Methods

Rushmeier [78] suggested using the zonal method (section 4.1.1) in a first pass to compute the source radiances. In order to deal with anisotropy for the single scattering approximation, a Monte Carlo second pass is used in the rendering step which accounts for one extra directional bounce.

Sakas [81] presents a method to render arbitrary distributions of volume densities by using projective polygonal rendering and solid texturing techniques. The general equation for the radiance reaching the eye for the single scattering case is derived, and can be solved with front-to-back Monte Carlo sampling. Two simplifications—special illumination geometries—are also presented: (1) for very low density volumes, absorption of the radiance coming from the light sources to the scattering points is neglected; (2) volume of constant density. A Bresenham algorithm is used to traverse the participating medium that is modeled as a voxel field (better schemes are presented in [83]), and created using fractal techniques.

Stam [92] introduces *stochastic rendering* of gaseous phenomena modeled as density fields, where the random element is transformed from the model to the rendering component. The statistics of the *intensity field* are related to the statistics of the phenomenon through the illumination equation. Instead of perturbing the model, the intensity is perturbed in a way that is consistent with both the model and the illumination equation. With a proper definition of an *average source radiance* $\bar{J}_{ss}$, the integral transport equation for single scattering (Equation 10) is rewritten as $L(x) = \tau(x_0,x)\,L(x_0) + (1 - \tau(x_0,x))\,\bar{J}_{ss}(x_0,x)$. The transmittance $\tau(x_0,x)$ is computed using the properties of the density field, and for the average source radiance $\bar{J}_{ss}(x_0,x)$ the algorithm of [95] is used.

3.4 Scientific visualization and volume rendering

Methods that solve the rendering equation under the assumption of single scattering can be related to volume rendering. Volume rendering, one of the visualization techniques in scientific visualization, is used to view 3D data directly. Voxels (volume elements) are used as representation of the volume to determine visual properties, such as opacity and color.

The main similarity between volume rendering and the rendering techiques that have been mentioned in the previous sections comes from the fact that the starting point of volume rendering is the so-called *volume rendering integral*, which is the same as the rendering equation. Nevertheless, there are important differences:

- The main distinction is that the objective of volume rendering is the generation of images that aid in the *comprehension* of a given volume model. Therefore, some assumptions can be done in volume rendering that are unacceptable in realistic rendering.
- In scientific visualization the unit volume that is rendered can be single valued or multivalued, representing a set of properties (temperature, pressure, oxygen concentration, humidity, etc.). The set of properties that the user is interested in are shown in the generated image and filters can be applied to select ranges of values of interest. In realistic rendering the radiance field is the single property of interest.

The volume rendering integral expresses the radiance variation through a light path, and that is what volume rendering methods must solve. From the different steps in the volume rendering visualization pipeline, only the *volume viewing* stage has a true connection with realistic rendering. Volume rendering methods are commonly classified into two categories: Object order (or *splatting*) methods, where the final pixel color accumulation is done by image composition, and image order methods where, generally, rays are cast from the observer through each pixel to compute the pixel colors via composition. Levoy [53] presents an expression to composite colors and opacities of the samples back-to-front along the line (for a certain wavelength):

$$C(x_i,y_j) = \sum_{k=0}^{n} \left[c(x_i,y_j,z_k)\,\alpha(x_i,y_j,z_k) \prod_{l=k+1}^{n} (1 - \alpha(x_i,y_j,z_l)) \right], \quad (11)$$

where $c(x_i,y_j,z_k)$ and $\alpha(x_i,y_j,z_k)$ are the "intensity" and opacity of k-th sample for the (x_i,y_j) ray, with $\alpha(x_i,y_j,z_0) = 1$ and $c(x_i,y_j,z_n)$ being the background color. Equation 11 can be shown to be a discretized version of the integral transport equation (Equation 4) with c representing the source radiance J and α being one minus the transmittance between consecutive samples. Also c can be considered to be J_{ss} (Equation 9) when neglecting multiple scattering. For example, Sakas and Hartig [84] use an illumination model accounting for single scattering for interactive viewing of large scalar voxel fields. Fig. 9 shows images obtained by ray casting scanned data.

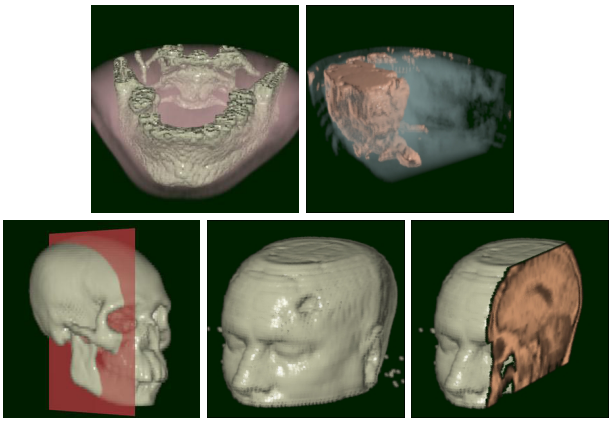

Fig. 9 Ray tracing scanned data (jaw, brain including an isosurface, and head). The images on the top row accumulate intensities along samples using a front-to-back style.

Kniss et al. [45], in the area of volume rendering visualization, have introduced a model of interactive volume shading which captures volumetric light attenuation effects. The model adds direct light attenuation to the classic volume rendering to incorporate volumetric shadows and improve visual perception, considers an approximation to phase functions, and to single and even multiple forward scattering. Their objective is not to perform a physically accurate computation of scattering effects, but the obtaining of interactive meaningful visualizations. Their model phenomenologically mimics scattering and light attenuation through the volume and takes advantage of graphics hardware.

3.5 Discussion

Most of the methods that neglect multiple scattering are focused on atmospheric effects, i.e. effects caused by the absorption or scattering of light in the air or in the particles present in the air. These methods make it possible to display various effects, such as the light beams caused by spotlights, shafts of light through clouds or within water, foggy scenes, smoke, the sky viewed from the Earth or the Earth viewed from space. A few of them focus on other media such as water [34] or fire [92]. Some proposed methods do not obtain very accurate results, as they are applied to situations where the single scattering approximation does not hold. (The single scattering approximation is only valid in the case of low albedo or optically thin media—Section 1.1.2—.)

A couple of these methods make strong simplifications not only about the medium, but also about the light sources and/or the viewer's position, making an analytical approximation possible [11, 106]. Nevertheless, in almost all applications these simplifications are unacceptable, and a more complex treatment is needed. Anyway, strong simplifications are made either regarding the medium optical properties or the sources. Most of the methods consider only homogeneous media [11, 54, 62, 51, 52], or planar layers of constant density [44, 54]. When rendering the Earth's atmosphere, spherical layers are common, either of constant [42], exponentially decreasing [42, 34] or linearly interpolated density [62]. Other researchers deal with voxels of constant properties [81] or even with properties varying from point to point [33].

Also the spectral dependency of scattering effects was not considered in the original methods [11, 41, 54, 62, 44] (with the exception of the work by Klassen [44]), although in order to reach certain accuracy a certain number of wavelengths—more than three—have to be used. This has been progressively recognized [17], and more recent methods deal with a sufficient set of wavelengths when necessary [98, 64, 34, 81].

Related to the light sources, it is common to work only with light sources placed at the infinity, such that the light angle is constant over the environment [11, 54, 44], but also with point light sources [54, 81, 62, 98] and with spotlights with angular distributions [62, 51, 52].

Even in the case of single scattering, the computational cost of calculating light scattering can be too expensive, specially when trying to apply it to drive or space/flight simulators where interactive rates are needed. Therefore, as in other areas, there is a general tendency to accelerate the rendering process by using graphics hardware. The intensities of the scattered light are computed in a pre-process and stored in look-up tables. These tables are used as 3D textures that are projected with bilinear interpolation to obtain the output images. This is for example the case of the work of Dobashi et al. [19] who have applied single scattering methods to different atmospheric effects to compute the appropriate textures, or the real-time animations of fog of Lecocq et al. [51, 52] and Biri et al. [8].

For the obtaining of realistic and visually appealing images, shadow consideration is essential. In the analytic method of Blinn [11], where the interaction between the light and the cloud of particles is simulated, shadowing is a result of blockage of light from other particles. If it is assumed that all particles have constant radius any statistical (in this case a Poisson process) can be used to model the probability that a particle is completely illuminated and that the reflected light being in the view direction does not intersect any other particle. Thus, an analytic solution to shadow determination is available. The ray-tracing method proposed by Kajiya and Von Herzen [41] is, evidently, able to consider both self-shadowing and shadows cast by the medium. To generate shadows for objects represented by densities they store the contribution of each light source to the brightness of each point in space into a 3D array. Ebert and Parent [22] improve the calculation of the shadow table by storing the shadow values already calculated in a 3D table and calculating the shadow table values starting with the point closest to the light and proceeding to the points farthest from the light to avoid repeated calculations. They also used a reduced-resolution shadow table, so that only a bilinear interpolation is needed to determine each value in the shadow table. However, the shadow algorithm using ray-tracing is very costly computationally. Anyway, as most deterministic single-scattering methods are oriented to solve specific problems, the majority of them do not explicitly consider shadows cast by the medium. This is the case of those centered

in the rendering of the atmosphere [44, 33, 42, 98, 64, 34, 19], of scenes containing fog [106, 54, 44, 62, 51] or of water effects [103, 63].

4 Multiple Scattering

Most of the methods that account for multiple scattering [70] use two stages: the *Illumination Pass*, in which the source radiance $J(x)$ —or other equivalent function— is computed (solving Equation 3), and the *Visualization Pass*, in which Equation 4 is solved for the points of the image plane.

We classify the existing methods into two main categories: deterministic and stochastic methods. Deterministic methods are further classified according to the space of directions, discerning between isotropic and anisotropic methods. All isotropic methods use constant basis functions for the computation of form factors. The very first of these methods is the zonal method [80], which is an extension to the classical radiosity method that accounts for isotropic emitting and scattering media. The zonal method has been improved by using hierarchies within the context of the progressive refinements method [6, 91] and also of hierarchical radiosity (HR) [69].

Deterministic methods can deal with anisotropy by means of spherical harmonics (P-N methods), discrete ordinates, or some implicit representation. Kajiya and Von Herzen [41] expand the radiance in a truncated spherical harmonic basis and construct a system of partial differential equations. Bhate and Tokuta [7] extend the zonal method by using a spherical harmonic basis. Discrete ordinates refers to the discretization of the direction space into a set of bins [65, 49, 55]. Using a grid of voxels to model the participating media, the transport equation can be solved locally per voxel, by means of *local interactions* [65, 49]. At each step the exiting radiance of a voxel is updated, consequently changing the incoming radiance of its neighbors, which must in turn solve their exiting radiances. Alternatively, using *global interactions*, the energy exchange between all pairs of elements can be considered, as an extension of the zonal method , setting and solving a system of equations whose coefficients are form factors. Max [55] approximates the effects of the form factors avoiding their computation.

Finally some deterministic methods use an implicit representation of the directional distribution of radiance (encoded either in scattering patterns [60], considering only a specific set of directions [59, 72, 38, 30, 31], by means of a diffusion equation [93, 94, 96] or of a point spread function [74, 75]). In the method of Nishita et al. [60], the contributions to the radiance $L(x)$ (for the second and third orders of scattering) in the viewing direction in a point x interior of a participating medium come in the form of a set of extended form factors in a grid that forms a 3D-filter. These form factors must be multiplied by the energy at the related points and accumulated to obtain $L(x)$. Second order scattering is considered in the works by Nishita et al. [59] and by Harris and Lastra [30, 31] for a highly reduced number of incoming directions.

Stam [93, 94, 96] uses a *diffusion approximation* to solve the multiple scattering between blobs modeling the media. Restricting the medium source radiance of a blob to be of the simple form $J_{\mathrm{m}}(\omega) = J^0 + \mathbf{J}^1 \cdot \omega$, a diffusion equation can be written as a system of linear equations allowing the calculation of the source radiance for each blob.

Stochastic methods solve the transport equation by means of random sampling, using random paths along interaction points. We distinguish between the methods that set the interaction points by using a constant step distance [9, 10], from those that sample a function of κ_{t}, which include light tracing [66, 68], bidirectional path-tracing [46], photon maps [37, 47, 23, 1] and Metropolis Light Transport [67]. Another categorization is made according to the view dependency of the methods. We tag a method as *view dependent* if it is image based or if in the Visualization Pass some extra process is needed to get the value of $L(x_0)$ (e.g. by using a ray tracing).

All these methods are summarized and categorized in Tables 3 and 4. Entries in italic style denote methods that do not solve the global illumination problem, in the sense that in the scene there is only a single volume to illuminate.

4.1 Deterministic Methods

4.1.1 Constant Basis Functions

Zonal Method The zonal method [80] is the extension to the classical radiosity method including isotropic participating media, which are modeled by voxels. The voxel radiosity is defined to include only the self-emitted plus the scattered energy (source function for a voxel). Form factors between volumes, and between volumes and surfaces are defined, and the form factors between surfaces are redefined to include a transmittance factor. They are computed by extending the hemicube technique. A system of s (for surfaces—patches) plus v (for volumes—voxels) related equations is constructed, and solved by the Gauss-Seidel iterative method. The direct application of the zonal method has a prohibitive cost: In a regular cube of n^3 voxels there are n^6 form factors; approximating them by the 1D integral along the centers of each pair of voxels in time $O(n)$ (i.e. number of intervening voxels) the computation of all form factors takes $O(n^7)$. Coherence between form factors has been exploited to compute them with lower cost [5].

Progressive Refinement Approach These methods [6,91] establish a fixed hierarchy in a preprocessing step and thereafter use it in a shooting strategy. There are no further refinements of the hierarchy that would enable the computation of a cheaper coarse solution that could be iteratively improved by refining it, as proposed in [29].

Bhate's method [6] is a progressive refinement version of the zonal method, using hierarchies. These hierarchies are computed in two preprocessing steps, which consist in the subdivision of volumes and surfaces and the creation of links between volumes and volumes and also between surfaces and volumes, determining the level at which a pair of elements must interact. Each global volume is refined against each other,

Table 3 Deterministic multiple scattering methods.

Space of directions					
Isotropic	*Anisotropic*				
Constant basis functions	*Spherical harmonics*	*Discrete ordinates*			*Implicit representation*
Zonal method: [80]	*[41]* [7]	*Based on local solutions (local interactions)*		*Global interactions*	3D-filter/N directions: *[60, 59, 72, 38, 30, 31]* [74, 75]
Hierarchy					
Progr. Ref.: [6, 91] / HR: [87, 69]		Progr. Ref.: *[65]*	Sweeps: [49] *[14]*	Sweeps: *[55]*	Diffusion: [93, 94, 96] *[26]*

Table 4 Stochastic multiple scattering methods.

	Distance sampling	
	Constant	*Random*
view independent	Light tracing: *[9]*	Light tracing: [66]
view dependent	Light tracing: [10]	Bidirectional path-tracing: [46, 20] Photon maps: [37, 47, 23, 1] Metropolis Light Transport: [67]

but not against itself (there is no self-refinement). The proposed heuristics for the volume-volume refinement are: Total form factor (when a rough estimate of the form factor is below some specified threshold, then the related elements can interact at the current level), estimated visibility between the volumes, and the optical depth of the intervening medium. Basically, in these last heuristics, when the transmittance between two elements is high, there is no need for further refinement. The volume-surface refinement also includes a brightness factor heuristic (for light sources). No Push-Pull procedure to set correctly the values of the radiosity at all levels of the hierarchies is performed.

In Sobierajski's method [91] the volumetric data is represented by voxels, that can model Lambertian surfaces, isotropic media, or a combination of both. Thus each voxel's BRDF is the sum of ideal diffuse reflection plus isotropic scattering. Depending on the specific coefficients for each component, a voxel can have a more translucent volumetric appearance or resemble more an opaque surface. Therefore, each voxel has a *diffuse* plus an *isotropic* radiosity. Form factors are defined to take into account the relationships between diffuse and isotropic components of the voxel's BRDF, and the surfaces. The presented technique is an iterative shooting algorithm using hierarchies which—for the case of volumes—are built in a preprocessing step by combining eight neighboring voxels at a certain level to form one voxel of the parent level. A criterion is defined to decide if a parent voxel can be a good approximation of its descendents. The interaction between nodes depends on their levels, their averaged values and the amount of energy transferred between them. There are no explicit links between nodes, instead at each shooting iteration the best highest possible levels of interaction are found *on-the-fly*. After each shooting iteration a Push-Pull procedure assures the correct representation of the energies of all the nodes in the hierarchies.

Hierarchical Radiosity Sillion [87] presents a hierarchical radiosity algorithm [29] adapted to include isotropic volumes. To represent energy exchanges within a volume, the self-link (link from the volume to itself) is introduced. This link is subdivided in a different way from links between different elements, since each child must include a self-link apart from the usual links between each pair of children. Furthermore, to avoid the quadratic cost of the initial link phase of the classical hierarchical method, that can be overwhelming in complex scenes, the transfer of energy between groups of objects (i.e. sets of surfaces and volumes) is allowed. These groups of objects compose abstract objects (clusters) that exchange energy as a whole. A hierarchy is created above the surface level, and then the initial linking phase is reduced to the creation of a single self-link from the top of the hierarchy to itself, representing the interactions taking place inside the global volume enclosing the scene. Once the initial link is refined by a recursive procedure, gathering and Push-Pull steps are performed until there is no significant change in the radiosities of any element. Care must be taken to perform correctly the Push-Pull procedure when dealing with inhomogeneous media and textured surfaces. Refinement of the links is done by bounding the radiosity transfer.

This hierarchical radiosity algorithm [87] was later extended by Pérez et al. [69], as a first pass of a two pass method, to account for anisotropically scattering participating media in addition to diffuse surfaces. This is accomplished by storing directional information as introduced by Sillion et al. [88] for clusters containing non-diffuse surfaces. Each cluster or participating medium element has associated a number of directional distributions which represent their radiant properties. Energy exchanges between all kinds of elements are treated in a uniform way. The extension takes into account the possible anisotropy of participating media by considering the phase function in the scattering at the leaves.

4.1.2 Spherical Harmonics Kajiya and Von Herzen [41] present two methods. The first one deals with single scattering, and the second with multiple scattering within the participating media. The radiance is expressed in a truncated spherical har-

monics basis, and a system of partial differential equations (PDEs) is constructed for the spherical harmonics coefficients. The system of PDEs is set and solved by relaxation. Only the constant phase function and the Rayleigh phase function are considered, and the expansion in spherical coordinates is truncated after the fourth coefficient (because "only the first few spherical harmonics are necessary for a convincing image", but obviously the cost of the method depends largely on the number of coefficients). The effects between surfaces and volumes are not taken into account.

The method by Bhate and Tokuta [7] deals with the effects between surfaces and volumes missed in [41], being an extension to the zonal method, in which the assumption of the isotropy of the medium is eliminated (through a representation of the phase function and radiance by using spherical harmonics) and the surfaces remain ideal diffuse. The phase function is approximated by the first M terms of its spherical harmonics expansion (approximation to Mie scattering). Note that a large number of form factors will to be computed, taking into account the spherical harmonics. These are calculated with the extended hemicube technique. Finally, a system of Mv equations for v volumes plus s related equations for s surfaces is set and solved using a Gauss-Seidel iterative technique. The direct application of this method is impractical because of its prohibitive cost: In a regular grid of $v = n^3$ voxels the cost to compute the form factors is $O(n^7 + M^2 n^6)$.

4.1.3 Discrete Ordinates Another possibility to account for directional functions is the use of discrete ordinates [86], i.e. a discretization of the full 4π solid angle into a set of bins. These represent particular directions, and it is supposed that, for sufficiently small volume elements, the properties are constant for each direction within each volume. The main problem of the discrete ordinates is the "ray effect" problem [50] since the energy is propagated through discrete directions instead of into the whole discrete solid angle.

Local Interactions Patmore [65] formulates the local solution of the transfer equation for the discrete directional model resulting of the subdivision of the volume (resulting in a cubic lattice) and the angular spaces (using in practice 6 or 26 directions). The participating medium considered is non-emitting, since the objective is to render clouds. A global solution of the transfer equation is obtained through iteratively obtaining local solutions (related to points of the cubic lattice). As a consequence of a local solution the unshot energies of the related point are updated. A new local solution is computed for the lattice point adjacent and in the direction of the highest unshot energy of the previous one, thus effectively following importance-based paths, until the unshot energy is below some threshold or the path exits the volume. This method computes directly the radiances exiting the volume, so no integration of source radiances are needed in the visualization pass.

The method by Languénou et al. [49, 48] follows a progressive refinement approach. The usual shooting method for surfaces is extended to account for the transmittance through the media, and also source terms within the media are updated accordingly. The radiosities of the boundaries of the media are computed by propagating the radiance (coming from the previously accumulated source terms) along all the discrete ordinates and using as many iterations as necessary to converge. Each iteration consists in a loop for each direction, in which a complete sweep of the voxel grid is performed to propagate the accumulated energy through adjacent voxels, starting from a convenient boundary voxel (related to the direction considered), where $O(Mv)$ is the cost per iteration (M being the number of direction bins). Finally, the radiance of the boundary faces of the medium is shot, using hemisphere interpolation. The whole process is repeated until convergence. The visualization pass computes the pixel radiances by using the source radiances of the voxels.

Cerezo and Serón [14] have extended the discrete ordinates method of [49] to include objects and sources inside the participating medium and to handle highly anisotropic phase functions. They also extend the original method to consider volumetric inelastic processes, particularly fluorescence. This method has been applied to the rendering of underwater scenes where the sea is treated as a participating medium characterized by real experimental medium parametrizations.

Global Interactions Max's method [55] is devoted to render clouds. The computation of the M^2v^2 form factors of the finite elements formulation is avoided by approximating their effects as the energy is propagated across the grid. For each bin, this propagation is made distributing the flux to the related neighbor voxels simultaneously for all voxels belonging to a layer, in time $O(v \log n)$, for v^2 interactions. The "ray effect" is reduced because the energy is propagated through the whole bin, not only through a single direction. The attenuation between two voxels is not accumulated along the straight line joining them, but along a set of possible propagation paths. The multiple scattering events produced within a single receiving element are accounted for. Since the time to scatter the received flux of a voxel to the direction bins is $O(Mv)$, the final cost per iteration is $O(Mv \log n + M^2 v)$. Thus when the number of iterations required to converge is small compared to v, this method is better than computing the whole set of form factors (with a cost of $O(v^2 n + v^2 M)$) and solving the resulting system.

4.1.4 Implicit Representation The directional distribution of radiance can be represented implicitly by a scattering pattern [60], by a diffusion equation [93,94,96] or a point spread function [74, 75]. We also include in this category the methods that consider a set of specific directions [59,72,38,30,31], which can be considered as a restriction of the 3D-filter used in [60].

3D-Filter/Concrete Directions Nishita et al. [60] propose a method to display clouds taking into account multiple scattering and skylight (light reaching the cloud due to the atmosphere's scattering plus the reflected light from the Earth's surface). Radiance from a cloud reaching the eye is computed

from the sunlight multiple scattered plus the skylight single scattered by the particles of the cloud. Of the multiple scattering of the sunlight only the three first orders of scatterings are considered (to save computation time), computing separately the single scattering. For the contribution of the second and third order of scattering to the radiance in the viewing direction, the space including the cloud is subdivided into voxels, with the viewing direction as a principal axis. Instead of computing form factors between each pair of voxels, a smaller space with the mean density of the cloud is set, and the contribution ratios to the radiance of the center voxel (in the viewing direction) from the other voxels are computed, taking into account the sunlight direction. This is the *contribution-ratio pattern*, or 3D-filter. Since the scattering in clouds is mainly forward, most of the energy scattered at a point will lie within a relatively small solid angle. Using this fact it is possible to compute faster the extended form factors, concentrating the effort in those voxels which will effectively contribute to the center voxel, for paths having one or two scatterings (and using an stochastic method to select those voxels). The filter is applied to the voxels in the whole space storing for each voxel the light scattered due to the second and third order scattering in the viewing direction. In [61] the authors apply the same method to render snow, which has a high albedo and looks white due to multiple scattering. The shape and density distribution of snow is modeled using metaballs (Fig. 10).

Fig. 10 Rendering snow taking multiple scattering into account (Nishita et al [61]).

Nishita et al. [59] treat the rendering of the sky, taking into account multiple scattering. Both Rayleigh and Mie scattering are considered, for the scattering due to air molecules and to large particles such as aerosols and water droplets, respectively. Nishita et al. extend the work by Kaneda et al. [42], using their model of the atmosphere but computing the atmosphere's multiple scattering and the light reflected from the ground. Accumulated optical thickness as viewed from the sun is computed and stored in a preprocess to speed-up calculations—using interpolation—for skylight rendering from a certain viewpoint. Although Nishita et al. suggest a method to compute third and higher order scattering, since the source radiance due to higher order of scattering decreases drastically, only the second order scattering is effectively computed. This is estimated by considering only eight precise directions around any particular sampling point of the viewing ray. These correspond to the sunlight direction, the perpendicular direction to the sunlight, the horizontal direction along the Earth surface and the zenith direction. Some of these directions hit the earth surface, and the light scattered at those intersection points (due to direct sunlight and also to skylight) is also taken into account.

Preetham et al. [72] have considered the rendering of outdoor scenes including sunlight and skylight, and the effects of *aerial perspective* (de-saturation and color shift of distant objects). Two inexpensive analytic models are developed for these two objectives, whose results have been verified against standard literature from atmospheric science. The basic idea is to compute the sky spectral radiance function for a set of sun positions and turbidities (for which the method of Nishita et al. [59] was used), and then fit a parametric function (Fig. 11). Wilkie et al. propose in [105] a simple analytical model for skylight polarisation to be added to Preetham's skylight model (and similar ones).

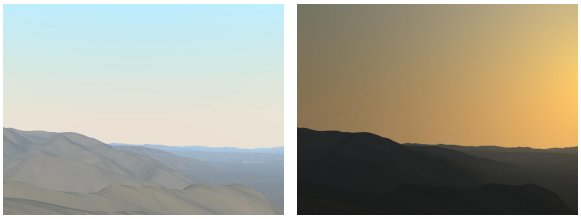

Fig. 11 Outdoor scene with physically-based sky model and aerial perspective (Preetham et al. [72]). The images have been rendered looking west at different times (left morning and right evening) with a turbidity value of 6.

Jensen et al. [38] have considered the rendering of nighttime sky including illumination coming from the Moon, the stars, the zodiacal light and the atmosphere. They use the atmospheric model and scattering phase functions of Nishita [59], and simulate transport with multiple scattering combining distribution ray-tracing and ray-marching: as the ray traverses the atmosphere, ray-marching is used to integrate the optical-depth sampling the in-scattered indirect radiance and the direct illumination in random positions.

The work by Harris and Lastra [30, 31] is concerned with the real-time rendering of high-quality static constant-shape clouds suitable for flight simulation and games (Fig. 12). It is based on a two-pass method by Dobashi et al. [18], where clouds were modeled using particle systems, isotropic multiple scattering was approximated by a constant ambient term, and 2D-textures were used for rendering. The method by Harris and Lastra extends that method with an approximation to multiple forward scattering (only in the light direction) and anisotropic first-order scattering (in the eye direction). It allows the viewer to fly in and around clouds and to see other flying objects passing through or behind them. Since the clouds are static and of constant shape, a per cloud preprocess can be done to compute the multiple forward scatter-

ing illumination. This is approximated by only considering the most significant direction from the whole sphere of directions: the light direction—this represents a harder restriction of the space of directions than the one done by Nishita et al. [60]. The anisotropic first order scattering is calculated at run-time. The Rayleigh phase function was used in this work, although indeed it is far from being well-suited for clouds—more accurate phase functions should be used for more accurate results. Particles are rendered using splatting [104]. In order to achieve real-time rendering in scenes containing many complex clouds, dynamically generated impostors are used (see e.g. [85]).

Fig. 12 Shading with only single scattering (left) and with multiple forward scattering (right). Anisotropic scattering simulation with two light sources, orange and pink, to simulate skylight (by Harris adn Lastra [30, 31]).

Diffusion Stam [93, 94] solves the global illumination by progressive refinements using shooting operations between patches, and between patches and blobs (which model the media). The shooting between blobs (that could be very expensive if the number of blobs v is large) is avoided by a set of v linear equations representing a diffusion equation. This is obtained by a *diffusion approximation* of the source radiance (due to the scattering of the medium radiance), i.e. it is characterized by only two functions: $J_{\mathrm{m}}(\omega) = J^0 + \mathbf{J}^1 \cdot \omega$. Solving the linear system allows the computation of the coefficients J^0 and $\mathbf{J}^1$ for each blob, and thus the multiple scattering between blobs. When v is not too large ($v < 1000$) the system can be solved with a direct LU-decomposition; for larger systems a relaxation scheme can be used, although the convergence is not guaranteed (but with a relatively small number of blobs good looking results are obtained). The proposed method uses far less memory and computation time than would be required by a grid method. Being a progressive method, when it deals with complex scenes composed of many surfaces, the cost of the progressive shooting of energy from the surface patches is quite expensive. A hierarchical approach would become necessary in such a case. In [96] Stam and Fiume apply the method to the animation of fire and other gaseous phenomena (Fig. 13).

Recently, Geist et al. [26], have presented a new technique based on the lattice-Boltzmann method able to deal with the difussion processes involved in participating medium illumination problems. They use a simple grid-based photon transport model able to handle complex boundary conditions. No interaction between light and surfaces is considered.

Fig. 13 Rendering smoke and fire using the diffusion approximation (by Stam and Fiume [96]).

Path integration Premoze et al [74] present an approximation to simulate the transport of light in volumes based on path integration. Unlike the diffusion approximation, that is only appropriate for dense uniform media, this approximation is valid for sparse and inhomogeneous media; nevertheless, they assume smoothly varying scattering coefficients and strongly-forward phase functions uniform throughout the medium. Their algorithm exploit the WKB (Wentzel-Kramers-Brillouin) approximation that computes the multiply-scattered light by finding the most probable path and then analytically integrates scattered radiance along this path and some neighborhood, including quadratic fluctuations around the path. They implement the algorithm using ray-marching. To speed up the direct component calculation, they pre-compute light attenuation in the volume and store it at all possible depths. The indirect component is computed constructing the most probable path and marching along the path in steps such that the number of scattering events is uniform. At every sampling point they compute the contribution from several directions over the sphere, but no rays are spawn. This approximation can be viewed as a complement to diffusion, as it can be applied to sparse and spatially-varying volumes where the diffusion approximation is not appropriate, but it ignores backscattering. They apply their method to the simulation of scattering in clouds and subsurface scattering. In a more recent paper, Premoze et al. [75] use again the mathematical path integration framework, but focusing this time on obtaining fast rendering algorithms, not numerically but qualitatively correct. They base their method on the observed fact that multiple scattering causes the blurring and attenuation of the incident radiance distribution. The idea is to avoid direct simulation of multiple scattering by taking advantage of the spatial spreading. To do this, they derive a point spread function from the path integration analysis of the light transport, function that

depends on the medium optical properties. In the preprocessing step they compute the attenuated light volume for each light source, but then blur the light volume with kernels of different sizes storing it at various levels of detail. Therefore, they reduce the calculation of the multiple scattering to a simple table look-up in the rendering stage.

4.1.5 Others Jackèl and Walter [35] focus on highly realistic atmospheric rendering, allowing the modeling of real world atmospheres (composed by ozone, haze, dust, soot, sulphur acid, etc.). The combination of four sub-models, consisting of a number of concentric shells, determine the atmospheric model: clear air, aerosol, ozone and rain layer. As in the work by Nishita et al. [59], both Rayleigh and Mie scattering are accounted for. They approximate the second order scattering to obtain more accurate results, but unfortunately the authors do not explain how this approximation is performed.

In the context of the fast rendering of atmospheric effects, Riley et al. [76] propose a multiple model system able to generate fine angularly dependent chromatic effects such as rainbows, halos and glories. The core of the system is an angular distribution renderer developped to handle the forward multiscattering behaviour of large water and ice atmospheric particles. It based on the use of multiple scattering angular distribution functions to approximate the distribution of light after a certain number of scattering events. It has been designed for efficient rendering in a single pass through the volume, taking advantage of programmable graphics hardware. The model applies the half angle slicing scheme for a light buffer and an eye buffer of Kniss el al. [45] (see section 3.4). This method slices volume data at an angle suitable for both eye and light perpectives (halfaway between them). Rendering is done both to an eye buffer to accumulate light directed into the eye rays and to a light buffer which stores the attenuation of light as it traverses the volume. Skylight and aerial perspective effects, based on a simplified single-scattering analytic model, are added in the rendering process to increase realism.

4.2 Stochastic Methods

Global illumination stochastic methods basically trace random rays within the environment. The interaction points that limit the rays can be obtained by using a constant step distance [9, 10] or sampling a cumulative density function [66, 46].

4.2.1 Constant Distance Sampling Blasi et al. [9, 10] describe methods to deal with participating media by using a simulation of the particle model of light (Monte Carlo light tracing). The first [9] deals with a single participating medium; the second [10] can render mixed scenes. Both take into account multiple scattering within the media, using the Schlick phase function, specially defined in such a way that the importance sampling is quite inexpensive, while maintaining the possibility of approximating other phase functions. In [9] it is used an approximation to the Mie scattering as a combination of isotropic plus forward scattering components. In the scattering events, the scatter direction is given by optimal importance sampling of the scattering component, and the isotropic part is stored in the voxel. This isotropic part of the voxel is not considered for the illumination of the other voxels. Since the directional component is much more important than the isotropic component, it is expected that the resulting error will not be significant. A progressive refinement strategy could be used when this isotropic energy becomes too important. Bundles progress in steps of constant length δ. Therefore, at each interaction point there is a sampling process to decide if there is scattering in that point. Absorption is taken into account along the whole path of the bundle, decreasing its flux at each step by the transmittance due to absorption along distance δ.

In [10] a progressive technique is used to render mixed scenes. Surfaces are classified as “diffuse” or “specular” depending on a threshold. In the illumination Pass, when a bundle hits a diffuse surface, its energy is stored there (and the bundle’s path ends), whereas when it hits a specular one, it is reflected using importance sampling. Within the media the bundles progress as explained above, although only when a bundle exits the media (if it does) its energy is recorded (at the border voxel). Due to this storage scheme, the number of rays travelling through the volumes must be higher than it would be required with a per voxel storage , to get an accurate sampling of the energy leaving the volume.

4.2.2 Random Distance Sampling

Light Tracing The Monte Carlo light tracing by Pattanaik and Mudur [66] uses a sampling process to find the points of interaction (absorption or scattering) of the bundles within the volume, with the expression $1 - \exp(-\int_0^S \kappa_t(u)\,du)$ as a cumulative distribution function, where S is the distance traveled. At those points, with the *Simple Absorption* method, another sampling process is performed to decide if the interaction is an absorption or a scattering event, based on Ω. On the other hand, with the *Absorption Suppression* method the bundles always scatter but they reduce their flux multiplying it by Ω. Different variance reduction techniques are proposed: Forced interaction of a bundle with each voxel, the just mentioned Absorption Suppression method , and the Particle Divergence method (in which the outgoing bundle is split into many bundles at the scattering points). The storage scheme presented is suited for isotropic scattering, but can be changed to deal with anisotropic scattering.

Roysam et al. [77] focus on the problem of visibility of lighted exit signs in buildings through fire-caused non-uniform smoke, proposing a parallelized Monte Carlo particle tracing method. A single step is performed in this method—just a particle tracing phase. Whenever a bundle hits the image plane (the receiver), the corresponding pixel is updated. Using a small number of bundles a coarse solution can be obtained. If the user requires a higher accuracy, the energy of

the already traced bundles is re-scaled and more bundles are shot. The scattering pattern corresponding to different smoke particles is measured experimentally and mixed into a tabulated phase function.

Bidirectional Path Tracing Lafortune and Willems present in [46] a bidirectional path tracing for non-emitting participating media (Fig. 14). Random walks are traced both from the light sources (light paths—light shooting) and from the eye point (eye paths—light gathering), being a combination of light tracing and eye tracing. Consequently this is an image-based method. After tracing a light path and an eye path, each intersection point of the respective paths are connected by shadow rays. Those shadow rays that are not occluded constitute a part of (complete) transport paths from the light sources to the eye, and an illumination contribution is computed for each transport path. These illumination contributions are combined to obtain an unbiased estimator for the radiance reaching the eye, taking into account the probability densities for generating the transport paths used. Concretely, the balance heuristic [100] is used to obtain the weights of the illumination contributions. Random walks (both light and eye rays) are traced computing interaction points within the media as in [66] (Simple Absorption case). For the scattering direction computation the Schlick phase function is used.

Fig. 14 Traffic scene without participating media (left) and with a foggy atmosphere(right), by Lafortune and Willems [46].

Dumont [20] is concerned with the design of solutions to improve road safety in foggy weather. A way to study the influence of fog effects is by means of simulations. The approach followed by Dumont is the use of a Monte Carlo algorithm for a homogeneous medium with constant phase function, extinction coefficient and albedo, being the theoretical equivalent values resulting from the combination of real data. The phase function is modeled by a goniometric distribution. In his work, Dumont resorts to the use of an extension (to deal with participating media) of the light tracing combined with next event estimation for the direct computation of pixel radiances, based on [21]. At each reflection or scattering event, the direct contribution to the camera (or to the cameras, if more than one is used) is calculated—thus, this method represents in fact a subset of bidirectional path tracing. This direct contribution is computed taking into account the PDF (Probability Density Function) of the event (to reflect/scatter in a particular direction) and the transmittance between the corresponding point and the camera.

Photon Maps Jensen and Christensen [37] remove previous restrictions limiting the photon map method to surfaces [36] introducing a *volume photon map* containing photons in the participating media (Fig. 15). This is a photon map different from the surface photon map because the way radiance is estimated from them is different—the authors also derive the radiance estimate expression for the volume photon map, using density estimation. The volume photon map is only used to represent indirect illumination, that is, it only stores photons that have been reflected or transmitted by surfaces before interacting with the media, and photons that have been scattered at least once in the media. It is created in a particle tracing preprocess, like in the work by Pattanaik and Mudur [66]. To account for anisotropic scattering, the direction of incidence is stored at hit points to recover in the rendering step the source radiance toward the eye.

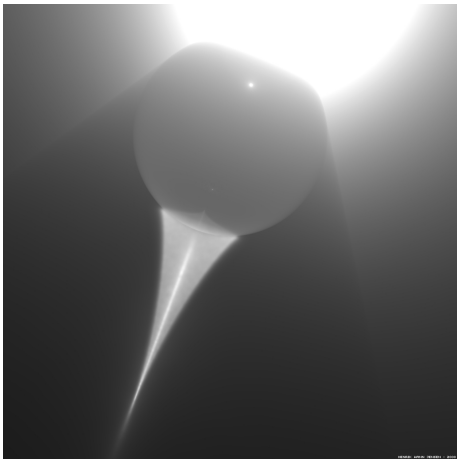

Fig. 15 Volume caustic in fog. Rendered using photon maps by Jensen and Christensen [37].

The same photon map method by Jensen and Christensen [37] was simultaneously suggested by Lange and Pietrek [47], although restricted to homogeneous isotropically scattering media—thus, in this case there is no need to store the incident direction in the photon map. Fedkiw et al. [23] use the method of [37] for visual simulation of smoke. In the second step, however, a forward instead of a backward ray marching algorithm is proposed, since it allows a more efficient culling of computations in smoke that is obscured by other smoke. Also a more efficient use of the photon map results by allowing to use less photons in the query as the ray gets deeper into the media.

Adabala and Manohar [1] use particle systems to model gaseous volumes, allowing the treatment of inhomogeneous anisotropically scattering media. By displacing the particles following fluids dynamics equations, the model can evolve in time, avoiding the use of grids. In order to manage the particles and their associated illumination field, a median kd-tree is used, dubbed by the authors *particle map*, storing information to be retrieved by associative searches. A particle tracing is used to distribute light within the media, in a way

that is similar to the photon map approach. However, when a scattering event takes place, instead of saving an entry in a photon map, the related information is stored in the particle map. The incident energy is distributed among particles near the point of interaction. Since the storage of the incident direction replicated onto those nearby particles would be prohibitive, only the corresponding energy that would be scattered into the viewing direction is stored, sacrificing the viewpoint independency of photon maps. The integral transport equation and the source radiance are solved using the particle maps by using nearest neighbor queries.

Metropolis Light Transport Pauly et al. [67] extend Metropolis Light Transport [101], which is based on the Metropolis sampling technique for handling difficult sampling problems in computational physics [57], to incorporate volumetric scattering (Fig 16). The resulting algorithm is unbiased, handles general geometric models and inhomogeneous media, accounts for multiple scattering, and uses little storage—at the expense of being a view dependent method. In order to render an image, a sequence of light transport paths are generated by randomly mutating single current paths (e.g. by adding a new vertex to the path). In the Metropolis technique the *path space* (set of all finite-length paths with an associated *path space measure*) is explored locally, favoring mutations that make small changes to the current path, focusing on the light paths that contribute most to the rendered image. Each path deposits a certain amount of energy to the pixel it passes through, updating the image. Therefore, the paths are distributed proportionally to their contribution to the final image. An initialization step determining the total image brightness is performed by means of a bidirectional path tracing execution.

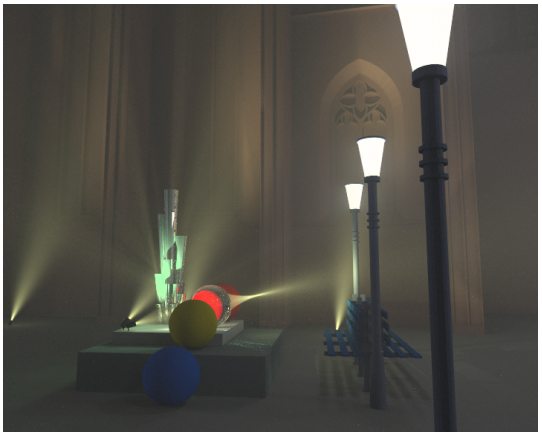

Fig. 16 Architectural model surrounded by a thin homogeneous medium simulating a foggy atmosphere (rendered with Metropolis Light Transport [67]).

4.3 Discussion

4.3.1 Applications As mentioned above, participating media functions (e.g. $\kappa_a(x)$, $\Omega(x)$, $p(\omega_o, \omega_i)$) can be extended for clusters. The computational cost of solving the global illumination problem in complex scenes can be dramatically reduced by the use of clusters, which represent both the transmission properties and the energy exchanges of their contained elements as a whole [88,90]. Thus, a suitable *extended* phase function for clusters could be used, for example, in a directional clustered hierarchical radiosity algorithm (see [16]).

It is clear that applications for safety analyses will need the best possible solution at any cost, starting from a sufficiently precise input model (for these, bidirectional path tracing could be used [46]), while for entertainment, for example, a visually pleasant image will be enough no matter if it is physically accurate (perhaps in this case isotropic media can be used, or the diffusion approximation ([93,94,96]). Training systems must perform real-time rendering of images, so the computation of images must be very fast, possibly reducing computation time by using visibly acceptable approximations and not too complex (dynamic) models. This invalidates the use of Monte Carlo based methods since they are highly time consuming. Extensions of the existing methods have to be studied for dealing with dynamic environments taking coherence into account, to achieve the necessary speed. Also the conditions of the particular problem to solve must be considered, such as the complexity of the scene (in terms of number of elements and their optical properties), types of image required, etc. For example, for sequence of a foggy driving simulation, the viewpoint is at a fixed distance from the ground and should follow a restricted path (must be over the road), on the other hand the *importance* of the elements of the scene must vary according to their position (relative to the viewpoint) in each frame.

4.3.2 Progressive Results The multi-gridding technique can be used to compute a sequence of solutions stopping when a sufficiently accurate solution is obtained. The sequence starts with a very rapid computation of a first coarse solution which is improved in successive steps. This can be accomplished in hierarchical approaches like in [87]. If a relatively small amount of time is given to compute a solution, then with the multi-gridding technique a coarse solution could be obtained.On the other hand, in the Monte Carlo light tracing method of Pattanaik and Mudur [66], the partial solution obtained in the Illumination Pass, for a given short time, will be far from the converged solution.This is due to the fact that each bundle follows its path in the scene until it dies. So, in practice, the image related to the partial results will not be useful.

In the light tracing method by Blasi et al. [10], however, the reflection on diffuse surfaces is eliminated, so that whenever a bundle hits a diffuse surface its path ends, and that diffuse surfaces accumulate unshot energy. At each iteration a set of bundles representing the unshot energy of the element having the highest value is sent to the environment, thus be-

ing a progressive refinement algorithm. Note that this technique also introduces bias since the process of reflection at a given point is substituted by a shooting from a random point within the element. The progressive nature of [10] allows the computation of an iterative sequence of images. However, the illumination of the media will be far from converged unless a very high number of bundles have been used. In the bidirectional ray tracing, since it is a view dependent method where the illumination is solved directly per pixel, the quality of the image can be gradually improved, starting from a very crude approximate image and converging to the solution as the program progresses.

Partial results of the illumination of a volume could be obtained between successive iterations in the methods which use discrete ordinates by sweeping of energy [49, 55]. Methods that do progressive refinements using hierarchies could do a good job if an ambient term is used for display purposes (as a generalization of the ambient term of the classical radiosity) after the shoot of light of the most energetic elements. This can also be used by other progressive refinements methods like the diffusion approximation [93, 94, 96], and the method by Blasi et al. [10].

4.3.3 Sampling Strategies It should be noted that the sampling strategy of [10] leads to biased results, while that of [66] does not. This is because the bundles can only be scattered at distances which are multiple of δ, and thus the expected length before scattering will not be equal to the mean free path without absorption. The error is reduced as the value of δ is diminished. Unfortunately to ensure results with a variance below some threshold the time required is approximately of the order of the inverse of δ. Moreover, we have checked that, for a same variance threshold and setting a value of δ relatively small to get a tolerable bias, the computation time using the sampling procedure of [66] is always lower than that of [10].

4.3.4 Isotropic Media It seems clear that for applications in which the isotropic assumption can be used (i.e. non-realistic applications), the obvious choice is the hierarchical radiosity method [87] since it has the best performance in computation time; moreover it is more reliable than the progressive refinements methods, in which a fixed hierarchy is used. Also the diffusion approximation using blobs method [93, 94, 96] could be simplified for the isotropic scattering case, being then a good choice (less memory and computation time than grid based methods) when the number of blobs is relatively small. When that number of blobs is high, then a hierarchical approach becomes necessary.

4.3.5 Anisotropic Media In the case of anisotropic media, all the existing methods commit errors; only the bidirectional path tracing, the photon maps approaches and Metropolis Light Transport are unbiased. It is then important to know what type of error can be accepted for each concrete application.

In the case of soft indirect illumination, the Metropolis method tends to use the same amount of computation time as brute force bidirectional path tracing. As Jensen and Christensen noticed [37], within participating media the illumination is mostly smooth because of the continuous scattering taking place everywhere in the medium. Therefore the benefits of Metropolis with respect to bidirectional path tracing can be marginal, depending on the concrete lighting conditions.

Stam [93, 94, 96] utilizes a *diffusion approximation*. This is only valid in the case of a high number of scattering events. This condition fails at the boundaries of the media, and thus the results are not so precise there; however, for certain applications (such as animations in which the objective is that things “look right”) they are accurate enough. Spherical Harmonics and Discrete Ordinates methods approximate directional functions by using a fixed set of bases. It could be interesting to use an adaptive number of bases in function of the accuracy required for the solution. Max [55] propagates the energy inside the medium in such a way that the “ray effect” (present in [65, 49]) is reduced, but computing an approximation of the true attenuation between two voxels; therefore although it produces visually better images it is not clear if that solution is less accurate than the direct Discrete Ordinates method (although for displaying clouds, for example, it is considered to be better). Obviously the bias of the Spherical Harmonics and Discrete Ordinates methods can be reduced by using a higher number of bases, but the computation time augments dramatically doing so. Thus there is a compromise between computation time and image quality. Comparing costs (see Fig. 17), we can sort the different methods starting from the cheapest as follows: Languénou et al discrete ordinates [49], Max’s discrete ordinates [55], direct extension of the zonal method with discrete ordinates, and the zonal spherical harmonics by Bhate and Tokuta [7]. However, it should be noticed that the accuracy of the results follows the reverse order.

The photon map approach introduced by Jensen and Christensen [37] is a simple and efficient method able to deal with complex geometries and lighting conditions. However, it requires high amount of memory for difficult lighting situations, and suffers from various artifacts, such as blurred shadows and caustic borders.

4.3.6 Shadows As it has been stated before, most multiple scattering methods split the problem into to stages: the Illumination phase, where the source function or an equivalent is computed, and the Visualization phase where the resulting image is calculated. Self-shadowing is always taken into account in the Illumination stage where the attenuation due to the medium is considered when calculating the volume illumination. Regarding the problem of casting shadows, it will depend on whether the method is able to solve the global illumination problem or not, ie, whether the resolution of the participating media is integrated into a global illumination system (entries not in italics in Tables 3 and 4) or not (entries in italics).

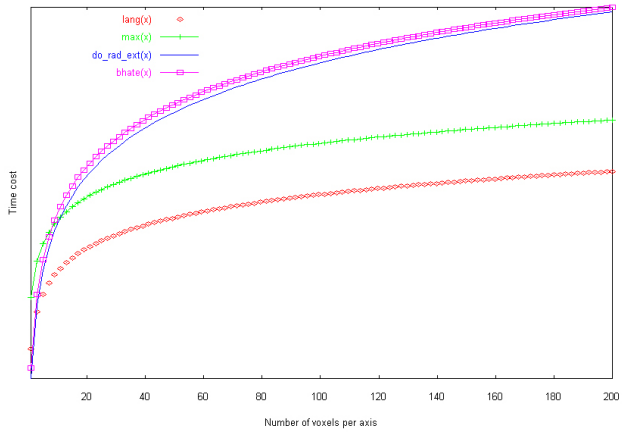

Fig. 17 Cost comparison: zonal spherical harmonics method of Bhate and Tokuta [7] (form factor computation: $O(n^7 + M^2n^6)$), the discrete ordinates methods of Languénou et al. [49] and Max [55] (iteration cost: $O(Mn^3)$ and $O(Mn^7 \log n + M^2n^7)$, respectively), and the direct extension of the zonal method with discrete ordinates ($O(n^7 + Mn^6)$, with $n^3 \gg M$). M represents either the number of terms in the spherical harmonics expansion, or the number of direction bins in the discrete ordinates methods, and n the number of voxels per grid axis. For the comparison the values used have been 96 bins and 15 iterations for the discrete ordinates methods, and 16 coefficients for the spherical harmonics expansion.

5 Reflections

5.1 Selecting the right existing method

The research results on the rendering of participating media are now widely used in a variety of applications such as stage lighting designs, entertainment, safety analysis, driving/space/flight simulators.

For some applications, like the entertainment ones, in which appearance is the key, fake media methods could be acceptable. However, if one is interested in physically-based methods, the single scattering ones cover the most simple, and thus not very accurate, techniques. The single scattering category groups together a variety of singular methods, including the first historical analytical ones. These analytical methods make so strong simplifications, not only about the medium's properties but the viewer's position and the light sources, that they are of no use in almost all applications. Nevertheless, within the single scattering category there are also more recent methods that focus on obtaining fast and pleasant participating media images. Most of the methods have been developed to solve problems related to specific media and scene configurations, and most of them deal with the rendering of atmospheric effects. It should be noted that the single scattering approximation is a strong simplification that is rarely fulfilled, but that is generally assumed for computational reasons (as is the case of the rendering of clouds). Even with the strong scattering simplification, the computational cost of obtaining the images is usually too high if interactive rates are needed. That makes it necessary, either to reconsider fake media methods or, as it has been lately investigated, to add new techniques using graphics hardware to speed up the rendering process.

If more general problems, dealing with complex scenes and participating media, have to be treated, multi-scattering methods have to be considered. If a fast computation of the images is needed, present Monte Carlo based methods are too time consuming. In the case of isotropic scattering (either real or assumed to speed up computations and lower memory requirements) the obvious choice is the hierarchical radiosity method since it has the best performance in computation time. Hierarchical approaches are also well suited to multigriding techniques, so that the sequence starts with a very rapid computation of a first coarse solution that can be improved in successive steps. Also the diffusion approximation using blobs could be a good choice since it requires less memory and computation time than grid based methods provided that the number of blobs is kept relatively small.

In the more complex and general case of anisotropic scattering, the most time and memory consuming methods are necessary. For applications (such as safety analyses) that require physically correct solutions at any cost, unbiased methods like the bidirectional path tracing, the Metropolis Light Transport and the photon maps approach have to be used. In the case of soft indirect illumination, the Metropolis method and the bidirectional path tracing are almost equally time consuming, slightly depending on the concrete lighting conditions. The photon map approach of Jensen and Christensen is a simple and efficient method able to deal with complex geometries and lighting conditions but it requires high amount of memory for difficult lighting situations.

Related to the other biased methods, it is important to know what type of error can be accepted for each specific application. The diffusion approximation is only valid in the case of a high number of scattering events, assumption that fails at the boundaries of the media so that biased results are obtained there. In any case, for certain applications the method can be precise enough and yields to "right looking" images. Spherical Harmonics and Discrete Ordinates methods are also approximated methods that use a fixed set of bases to deal with directionality. Obviously the higher the number of bases the lower the bias, but computation time and memory requirements dramatically increase. Thus, there should be a compromise between computation effort and image quality.

5.2 Future research directions

The different methods are the result of the struggle to treat lighting effects in a precise way to simulate underlying physical phenomena accurately and efficiently. The quality and completeness of the physical and mathematical models used to simulate every aspect of the light-participating media interaction are clear, but more work is needed regarding the efficiency of the calculations. In spite of spectacular improvements, the computational costs and memory requirements of all the advanced algorithms are quite high. An important line

of research is to find the best simplifications for specific problems to achieve the desired efficiency. A second, more general option is the intensive use of available hardware. According to Moore's Law, the acceleration rate is about 10 times every 5 years, so real-time (12 frames/s) computation of global illumination in scenes is about 50 years away. It may even take longer since Moore's Law may fail in the long run, for a number of reasons. However, several of the presented methods are very concurrent and well-suited for parallel implementation. Three different hardware platforms are now available:

- Special-purpose hardware architectures with VLSI chips explicitly designed for maximum performance [2].
- Programmable Graphics Processing Unit (GPU) used as massively parallel, powerful streaming processors [97,102].
- CPUs used to run highly optimized and parallel software algorithm implementations [24].

Real-time rendering based on graphics acceleration hardware is one of the most active areas of computer graphics [4]. The improvements in hardware can be roughly described in several stages. The first generation was wireframe displays, then shaded solids, texture mapping and a fourth stage beginning in 2001 with the programmable pipeline. The speed of the new modern accelerator, the Graphics Processing Unit (GPU), is doubled every year and should continue to do so. This represents an increase of about 50 every 5 years, compared to 10 according to Moore's Law. Kurt Akeley [2], one of the founders of SGI and now at Stanford and NVIDIA, predicts that the fifth generation of graphics hardware will tackle hand-on such issues as shadows, ray tracing, and other global illumination techniques. However, the GPU systems have significant limitations when dealing with realistically complex scenes with participating media. This is mainly why the global illumination algorithms have traditionally been used for off-line computations.

These problems can be solved using scalable massively parallel or distributed computing environment systems [3, 43]. Several sophisticated methods have been developed for efficient parallel simulation of illumination effects in complex environments. Recently, a number of researchers have implemented global illumination algorithms using the programmable features of graphics hardware (GPU) as a supercomputing co-processor [102]. However, they must be carefully planned to overcome the inherent difficulties of parallel processing. The original rendering problem has to be divided into sub-tasks, efficiently distributed among the processors to balance the processing load and the communication between subtasks for faster processing should be reduced. The traditional off-line parallel processing has significant limitations for realistic complex scenes containing millions of surfaces, thousands of light sources, and a high degree of occlusion.

Although there appears to be a mutual benefit from combining parallel computing and physically based rendering, surprisingly little work has been done to promote research in this field. Our hope is that this final reflection, which does not intend to be a comprehensive state of the art of the issue, will help to understand the possibilities of parallel processing on supercomputers, PC clusters, or GPUs, making parallel processing a more realistic and feasible option to approach new challenges.

Acknowledgments

We would like to thank the different authors for providing the images included in the survey.

This work has been partly financed by the Spanish "Comisión Interministerial de Ciencia y Tecnología" (contracts number TIC2001-2392-C03-02 and TIN2004-07672-C03-03) and the DURSI of the Generalitat de Catalunya (grant number 2001/SGR/00296).

References

1. N. Adabala and S. Manohar. Modeling and rendering of gaseous phenomena using particle maps. *Journal of Visualization and Computer Animation*, 11(5):279–294, 2000.
2. K. Akeley and P. Hanrahan. Real Time Graphics Architectures. Technical report, University of Stanford, 2001. Course CS448A Notes, Fall 2001. http://graphics-stanford.edu/courses/cs448a-01-fall.
3. T. Akenine-Moller and E. Haines. *Practical Parallel Rendering*. AKPeters, 2002. ISBN 1-56881-179-9.
4. T. Akenine-Moller and E. Haines. *Real-Time Rendering*. AKPeters, 2003. ISBN 1-56881-182-9.
5. D. Arquès and S. Michelin. Proximity Radiosity: Exploiting Coherence to Accelerate Form Factor Computations. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 143–152, New York, NY, 1996. Springer-Verlag/Wien.
6. N. Bhate. Application of Rapid Hierarchical Radiosity to Participating Media. In *Proceedings of ATARV-93: Advanced Techniques in Animation, Rendering, and Visualization*, pages 43–53, Ankara, Turkey, July 1993. Bilkent University.
7. N. Bhate and A. Tokuta. Photorealistic Volume Rendering of Media with Directional Scattering. In *Third Eurographics Workshop on Rendering*, pages 227–245, Bristol, UK, May 1992.
8. V. Biri, S. Michelin, and D. Arquès. Real-time animation of realistic fog. In *Rendering Techniques 2002 (Proceedings of the Thirteenth Eurographics Workshop on Rendering)*. ACM Press, June 2002. Poster paper. To appear.
9. P. Blasi, B. Le Saëc, and C. Schlick. A Rendering Algorithm for Discrete Volume Density Objects. *Computer Graphics Forum (Eurographics '93)*, 12(3):C201–C210, September 1993.
10. P. Blasi, B. Le Saëc, and C. Schlick. An Importance Driven Monte-Carlo Solution to the Global Illumination Problem. In *Fifth Eurographics Workshop on Rendering*, pages 173–183, Darmstadt, Germany, June 1994.
11. J. F. Blinn. Ligth Reflection Functions for Simulation of Clouds and Dusty Surfaces. *Computer Graphics (ACM SIGGRAPH '82 Proceedings)*, 16(3):21–29, 1982.
12. C. F. Bohren. Multiple Scattering of Light and Some of its Observable Consequences. *American Journal of Physics*, 55(6):524–533, 1987.
13. C. F. Bohren and D. R. Huffman. *Absorption and Scattering of Light by Small Particles*. John Wiley & Sons, 1993.

14. E. Cerezo and F. J. Serón. Rendering natural water: Merging computer graphics with physics and biology. In J. Vince and R. Earnshaw, editors, *Advances in Modelling, Animation and Rendering (Proc. of Computer Graphics International 2002)*. Springer, 2002.
15. S. Chandrasekhar. *Radiative Tranfer*. Dover, New York, 1960.
16. P. H. Christensen. *Hierarchical Techniques for Glossy Global Illumination*. PhD thesis, Seattle, Washington, 1995.
17. K. Devlin, A. Chalmers, A. Wilkie, and W. Purgathofer. Tone Reproduction and physically based spectral rendering. In *Eurographics 2002 State of the Art Reports, Saarbrücken (Germany), 2-6 September*, 2002.
18. Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita. A simple, efficient method for realistic animation of clouds. *Proceedings of SIGGRAPH 2000*, pages 19–28, July 2000. ISBN 1-58113-208-5.
19. Y. Dobashi, T. Yamamoto, and T. Nishita. Interactive rendering of atmospheric scattering effects using graphics hardware. In S. N. Spencer, editor, *Proceedings of the 17th Eurographics/SIGGRAPH workshop on graphics hardware (EGGH-02)*, pages 99–108, New York, 2002. ACM Press.
20. E. Dumont. Semi-monte carlo light tracing applied to the study of road visibility in fog. In *Proceedings of the Third International Conference on Monte Carlo and Quasi Monte Carlo Methods in Scientific Computing*, Lecture Notes in Computational Science and Engineering, Berlin, Germany, 1998. Springer Verlag.
21. P. Dutré, E. Lafortune, and Y. D. Willems. Monte Carlo Light Tracing with Direct Pixel Contributions. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 128–137, Alvor, Portugal, December 1993.
22. D. S. Ebert and R. E. Parent. Rendering and animation of gaseous phenomena by combining fast volume and scanline a-buffer techniques. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, pages 357–366, August 1990.
23. R. Fedkiw, J. Stam, and H. W. Jensen. Visual simulation of smoke. In *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 2001)*, pages 15–22, August 2001.
24. I. Foster. *Designing and Building Parallel Programs. Concepts and Tools for Parallel Software Engineering*. Addison Wesley, 1995.
25. G. Y. Gardner. Visual Simulation of Clouds. *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, 19(3):297–303, July 1985.
26. R. Geist, K. Rasche, J. Westall, and R. Schalkoff. Lattice-boltzmann lighting. In *Eurographics Symposium on Rendering*, pages 355–362, Norrköping, Sweden, 2004.
27. A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.
28. P. Hanrahan and W. Krueger. Reflection from Layered Surfaces due to Subsurface Scattering. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 165–174, 1993.
29. P. Hanrahan, D. Salzman, and L. Aupperle. A Rapid Hierarchical Radiosity Algorithm. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.
30. M. J. Harris and A. Lastra. Real-time cloud rendering. In *Computer Graphics Forum (Proceedings of Eurographics 2001)*, volume 20, pages 76–84, September 2001. Available on Computer Graphics Forum Volume 20 CD-ROM.
31. M. J. Harris and A. Lastra. Real-time cloud rendering for games. In *Proceedings of Game Developers Conference 2002*, March 2002.
32. H. C. v. d. Hulst. *Light Scattering by Small Particles*. Dover Publications, New York, NY, 1981.
33. M. Inakage. An Illumination Model for Atmospheric Environments. In R. A. Earnshaw and B. Wyvill, editors, *New Advances in Computer Graphics*, pages 533–548. Springer-Verlag, New York, NY, 1989.
34. J. Irwin. Full-Spectral Rendering of the Earth's Atmosphere Using a Physical Model of Rayleigh Scattering. In *Proceedings of the 1996 Eurographics UK Conference*, pages 103–115, 1996.
35. D. Jackèl and B. Walter. Modeling and Rendering of the Atmosphere Using Mie-Scattering. *Computer Graphics Forum*, 16(4):201–210, 1997.
36. H. W. Jensen. Global Illumination Using Photon Maps. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 21–30, New York, NY, 1996. Springer-Verlag/Wien.
37. H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Computer Graphics (ACM SIGGRAPH '98 Proceedings)*, pages 311–320, 1998.
38. H. W. Jensen, F. Durand, M. Stark, S. Premoze, J. Dorsey, and P. Shirley. A physically-based night sky model. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '01)*, pages 399–408, August 2001.
39. H. W. Jensen, J. Legakis, and J. Dorsey. Rendering of wet materials. In D. Lischinski and G. W. Larson, editors, *Rendering Techniques '99 (Proceedings of the Tenth Eurographics Workshop on Rendering)*, pages 273–282, New York, NY, 1999. Springer Wien.
40. H. W. Jensen, S. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '01)*, pages 511–518, August 2001.
41. J. T. Kajiya and B. P. V. Herzen. Ray Tracing Volume Densities. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 165–174, July 1984.
42. K. Kaneda, T. Okamoto, E. Nakamae, and T. Nishita. Photorealistic Image Synthesis for Outdoor Scenery Under Various Atmospheric Conditions. *Visual Computer*, 7(5):247–258, 1991.
43. T. Kato and J. Saito. The kilauea Parallel Global Illumination Renderer. In *Rendering Techniques '96 (Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization)*, pages 7–16, New York, NY, 2002. Springer-Verlag/Wien.
44. R. V. Klassen. Modeling the Effect of Atmosphere on Light. *ACM Transactions on Graphics*, 6(3):215–237, 1987.
45. J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, 2003.
46. E. P. Lafortune and Y. D. Willems. Rendering Participating Media with Bidirectional Path Tracing. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 91–100, New York, NY, 1996. Springer-Verlag/Wien.
47. T. Lange and G. Pietrek. Rendering Participating Media Using the Photon Map. Technical Report 678/1998, Fachbereich Informatik, Universität Dortmund, 1998.

48. E. Languénou. *Radiosité Hiérarchique et Transfer Radiatif Dans Les Milieux Semi-Transparents*. PhD thesis, Universite de Rennes, October 1994. Available (in French) from http://www.sciences.univ-nantes.fr/info/perso/permanents/languenou/renderWork.html.
49. E. Languénou, K. Bouatouch, and M. Chelle. Global Illumination in Presence of Participating Media with General Properties. In *Fifth Eurographics Workshop on Rendering*, pages 69–85, Darmstadt, Germany, June 1994.
50. K. D. Lathrop. Ray effects in discrete ordinates equations. *Nuclear Science and Engineering*, 32:357–369, 1968.
51. P. Lecocq, S. Michelin, D. Arquès, and A. Kemeny. Simulation d'éclairage en présence de milieux participatifs : Vers une solution temps-réel. *AFIG '00 (Actes des 13ièmes journées de l'AFIG)*, December 2000.
52. P. Lecocq, S. Michelin, A. Kemeny, and D. Arquès. Real time lighting simulation in presence of fog: Applications for driving simulation. In *Proceedings of the Driving Simulation Conference 2002*, Paris , France, September 2002.
53. M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.
54. N. L. Max. Atmospheric Illumination and Shadows. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, pages 117–124, August 1986.
55. N. L. Max. Efficient Light Propagation for Multiple Anisotropic Volume Scattering. In *Fifth Eurographics Workshop on Rendering*, pages 87–104, Darmstadt, Germany, June 1994.
56. N. L. Max. Optical models for direct volume rendering. *IEEE Transactions*, 1(2):99–108, Jun 1995.
57. N. C. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
58. D. Q. Nguyen, R. P. Fedkiw, and H. W. Jensen. Physically based modeling and animation of fire. *ACM Transactions on Graphics*, 21(3):721–728, 2002.
59. T. Nishita, Y. Dobashi, K. Kaneda, and H. Yamashita. Display Method of the Sky Color Taking Into Account Multiple Scattering. In *Proceedings of the Fourth Pacific Conference on Computer Graphics and Applications (Pacific Graphics '96)*, pages 66–79, 1996.
60. T. Nishita, Y. Dobashi, and E. Nakamae. Display of Clouds Taking Into Account Multiple Anisotropic Scattering and Sky Light. In *Computer Graphics Proceedings, Annual Conference Series, 1996 (ACM SIGGRAPH '96 Proceedings)*, pages 379–386, 1996.
61. T. Nishita, H. Iwasaki, Y. Dobashi, and E. Nakamae. A modeling and rendering method for snow by using metaballs. *Computer Graphics Forum (Eurographics '97 Proceedings)*, 16(3), September 1997.
62. T. Nishita, Y. Miyawaki, and E. Nakamae. A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources. In *Computer Graphics (ACM SIGGRAPH '87 Proceedings)*, volume 21, pages 303–310, July 1987.
63. T. Nishita and E. Nakamae. Method of Displaying Optical Effects within Water using Accumulation Buffer. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 373–380, 1994.
64. T. Nishita, T. Sirai, K. Tadamura, and E. Nakamae. Display of the Earth Taking Into Account Atmospheric Scattering. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 175–182, 1993.
65. C. Patmore. Simulated Multiple Scattering for Cloud Rendering. In S. P. Mudur and S. N. Pattanaik, editors, *Graphics, Design and Visualization (IFIP Transactions B-9)*, pages 59–70, Amsterdam, The Netherlands, 1993. North-Holland.
66. S. N. Pattanaik and S. P. Mudur. Computation of Global Illumination in a Participating Medium by Monte Carlo Simulation. *The Journal of Visualization and Computer Animation*, 4(3):133–152, July - September 1993.
67. M. Pauly, T. Kollig, and A. Keller. Metropolis light transport for participating media. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 11–22, New York, NY, 2000. Springer Wien.
68. F. Pérez, I. Martín, and X. Pueyo. High Quality Final Gathering for Hierarchical Monte Carlo Radiosity for General Environments. In J. Vince and R. Earnshaw, editors, *Advances in Modelling, Animation and Rendering (Proc. of Computer Graphics International 2002)*, pages 425–437. Springer, 2002.
69. F. Pérez, I. M. n, F. X. Sillion, and X. Pueyo. Acceleration of monte carlo path tracing in general environments. In *Proceedings of Pacific Graphics 2000*, Hong Kong, PRC, October 2000.
70. F. Pérez, X. Pueyo, and F. X. Sillion. Global illumination techniques for the simulation of participating media. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 309–320, New York, NY, 1997. Springer Wien.
71. K. Perlin. An Image Synthesizer. *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, 19(3):287–296, August 1985.
72. A. J. Preetham, P. Shirley, and B. E. Smits. A practical analytic model for daylight. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '99)*, pages 91–100, August 1999.
73. S. Premoze and M. Ashikhmin. Rendering natural waters. *Computer Graphics Forum*, 20(4):189–200, 2001.
74. S. Premoze, M. Ashikhmin, and P. Shirley. Path integration for light transport in volumes. In *Proceedings of Eurographics Symposium on Rendering*, pages 52–63, June 2003.
75. S. Premoze, M. Ashikhmin, J. Tessendorf, R. Ramamoorthi, and S. Nayar. Practical rendering of multiple scattering effects in participating media. In *Eurographics Symposium on Rendering*, Norrköping, Sweden, 2004.
76. K. Riley, D. Ebert, M. Kraus, J. Tessendorf, and C. Hansen. Efficient rendering of atmospheric phenomena. In *Eurographics Symposium on Rendering*, pages 375–386, Norrköping, Sweden, 2004.
77. B. Roysam, A. Cohen, P. Getto, and P. Boyce. A Numerical Approach to the Computation of Light Propagation Through Turbid Media: Application to the Evaluation of Lighted Exit Signs. In *IEEE Conference on Industry Applications*, pages 661–669, Detroit, Mich., 1993.
78. H. E. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. PhD thesis, Ithaca, NY, 1988.
79. H. E. Rushmeier. Rendering Participating Media: Problems and Solutions from Application Areas. In *Fifth Eurographics Workshop on Rendering*, pages 35–56, Darmstadt, Germany, June 1994.

80. H. E. Rushmeier and K. E. Torrance. The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium. In *Computer Graphics (ACM SIGGRAPH '87 Proceedings)*, volume 21, pages 293–302, July 1987.
81. G. Sakas. Fast rendering of arbitrary distributed volume densities. In C. E. Vandoni and D. A. Duce, editors, *Eurographics '90*, pages 519–530, Amsterdam, Netherlands, 1990. North-Holland. ISBN 0-444-88683-4.
82. G. Sakas. Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer*, 9(4):200–212, January 1993.
83. G. Sakas and M. Gerth. Sampling and anti-aliasing of discrete 3-D volume density textures. In *Eurographics '91*, pages 87–102, 527, Amsterdam, North-Holland, September 1991. Elsevier Science Publishers.
84. G. Sakas and J. Hartig. Interactive visualization of large scalar voxel fields. In A. Kaufman and G. M. Nielson, editors, *Proceedings Visualization '92*, Boston, Massachusets, October 1992.
85. G. Schaufler. Dynamically generated impostors. *GI Workshop on Modeling, Virtual Worlds, Distributed Graphics, Bonn, Germany*, pages 129–139, 1995.
86. R. Siegel and J. R. Howell. *Thermal Radiation Heat Transfer, 3rd Edition*. Hemisphere Publishing Corporation, New York, NY, 1992.
87. F. Sillion. A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995.
88. F. Sillion, G. Drettakis, and C. Soler. A Clustering Algorithm for Radiance Calculation in General Environments. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 196–205, New York, NY, 1995. Springer-Verlag.
89. F. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, CA, 1994.
90. B. Smits, J. Arvo, and D. Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 435–442, 1994.
91. L. M. Sobierajski. *Global Illumination Models for Volume Rendering*. PhD thesis, Stony Brook, NY, August 1994.
92. J. Stam. Stochastic Rendering of Density Fields. In *Proceedings of Graphics Interface '94*, pages 51–58, San Francisco, CA, May 1994. Morgan Kaufmann.
93. J. Stam. *Multi-Scale Stochastic Modelling of Complex Natural Phenomena*. PhD thesis, University of Toronto, Dept. of Computer Science, 1995. http://visinfo.zib.de/EVlib/Show?EVL-1995-169.
94. J. Stam. Multiple Scattering as a Diffusion Process. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 41–50, New York, NY, 1995. Springer-Verlag.
95. J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 369–376, 1993.
96. J. Stam and E. Fiume. Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 129–136, 1995.
97. W. M. T. Purcell, I. Buck and P. Hanrahan. Ray tracing on programmable Graphics Hardware. *ACM Transactions on Graphics*, 21(3):703–712, July - September 2002.
98. K. Tadamura, E. Nakamae, K. Kaneda, M. Baba, H. Yamashita, and T. Nishita. Modeling of Skylight and Rendering of Outdoor Scenes. In *Computer Graphics Forum (Eurographics '93)*, volume 12, pages C189–C200, Barcelona, Spain, September 1993.
99. F. Uhl and J. Blanc-Talon. Rendering Explosions. In *SCS/SMC 97*, April 1997.
100. E. Veach and L. J. Guibas. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Computer Graphics Proceedings, Annual Conference Series, 1995 (ACM SIGGRAPH '95 Proceedings)*, pages 419–428, 1995.
101. E. Veach and L. J. Guibas. Metropolis light transport. In *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, volume 31, pages 65–76, 1997.
102. P. T. S. J. B. C. S. P. Wald, I. Realtime Ray Tracing and its use for interactive Global Illumination. Technical report, Computer Graphics Group of the Saarland University, Germany, 2003. Eurographics State of the Art Reports, http://www.openrt.de/index.html.
103. M. Watt. Light-Water Interaction using Backward Beam Tracing. *Computer Graphics*, 24(4):377–385, 1999.
104. L. Westover. Footprint evaluation for volume rendering. *Computer Graphics*, 24(4):367–376, August 1990.
105. A. Wilkie, R. Tobler, C. Ulbricht, G.Zotti, and W. Purgathofer. An analytical model for skylight polarisation. In *Eurographics Symposium on Rendering*, Norrköping, Sweden, 2004.
106. P. J. Willis. Visual simulation of atmospheric haze. *Computer Graphics Forum*, 6:35–42, 1987.
107. L. Yaeger, C. Upson, and R. Myers. Combining Physical and Visual Simulation - Creation of the Planet Jupiter for the Film 2010. *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, 20(4):85–93, August 1986.

Biographies

Eva Cerezo is an associated lecturer at the University of Zaragoza, Spain, and a member of the Advanced Computer Graphics Group. She has received a PhD in Computer Science in 2002, a MsC in Nuclear Physics in 1992 and a BS in Physics in 1990 from the University of Zaragoza. Her research interests include the simulation of natural phenomena and behavior-based modeling.

Frederic Pérez obtained a BS in Computer Science from the Universitat Politècnica de Catalunya in 1992 and a PhD in Computer Science in 2003. His research interest include image processing in medical imaging, high quality rendering and global illumination techniques. He is currently employed as a researcher by the Foundation Research Institute of the Vall d'Hebron Hospital, collaborating with the Digestive System Research Group.

Xavier Pueyo obtained the PhD degree in Industrial Engineering from Universitat Politécnica de Catalunya (UPC) in 1986 and the Docteur Ingénieur degree from Université de Rennes I (France) in 1984. He is a professor in the Department of Computer Science and Applied Mathematics of the Universitat de Girona (UdG), Spain. He has chaired the Eurographics Working Group on Rendering, the Workshops and Working Groups Board of Eurographics and the Spanish Chapter of Eurographics. He has led several national and international research projects in the area of computer graphics and he has published in the fields of visibility computation and realistic rendering.

Francisco J. Seron is professor of Computer Science at the Technical School of Engineering at the University of Zaragoza. He received a Physical Science Degree from the University of Zaragoza in 1977 and a Ph.D. from the same University in 1984. At present he is the head of the Advanced Computer Graphics Group within the Computer Science Department. His research interests include Simulation of Natural Phenomena, Illumination Engineering and Virtual Reality.

Francois X. Sillion is currently a senior researcher at INRIA (Institut National de Recherche en Informatique et Automatique). He is the head of the ARTIS research group, and scientific advisor of INRIA Rhône-Alpes. He is also an associate professor at Ecole Polytechnique. He graduated from Ecole Normale Supérieure in Paris in 1988 and obtained a PhD from Université Paris-XI in 1989. His research interests include: simplification and detail addition to 3D models, visualization of very large data, tool development for expressive rendering, lighting simulation for image synthesis and data acquisition from real images.