Local and Global Illumination in the Volume Rendering Integral

Nelson Max[1] **and Min Chen**[2]

1 **University of California, Davis, USA**
max@cs.ucdavis.edu

2 **Swansea University, Swansea, UK**
m.chen@swansea.ac.uk

Abstract

This article is intended as an update of the major survey by Max [40] on optical models for direct volume rendering. It provides a brief overview of the subject scope covered by [40], and brings recent developments, such as new shadow algorithms and refraction rendering, into the perspective. In particular, we examine three fundamentals aspects of direct volume rendering, namely *the volume rendering integral*, *local illumination models* and *global illumination models*, in a wavelength-independent manner. We review the developments on *spectral volume rendering*, in which visible light are considered as a form of electromagnetic radiation, optical models are implemented in conjunction with representations of spectral power distribution. This survey can provide a basis for, and encourage, new efforts for developing and using complex illumination models to achieve better realism and perception through optical correctness.

1998 ACM Subject Classification I.3.7 Three-Dimensional Graphics and Realism

Keywords and phrases Volume Rendering, Illumination Model

Digital Object Identifier 10.4230/DFU.SciViz.2010.259

1 Introduction

The process of volume rendering maps a 3D scalar field into a 3D field of optical properties, usually color c and extinction coefficient τ, via so-called transfer functions, and then approximates its visual appearance by integrating along viewing rays.

Max provided a comprehensive survey of optical models for direct volume rendering in 1995 [40], and the survey has been extensively referenced in the volume rendering literature. Since then, there have been some new developments in the context of optical and illumination models for volume rendering, such as spectral volume rendering, shadow algorithms and refraction rendering. The authors believe that it is beneficial to re-visit this topic and bring these new developments into the perspective. On one hand, this survey is written as an update of [40], so it does not repeat the details of many technical aspects that were presented in [40]. On the other hand, it is also intended to be a self-contained work, so some fundamentals have been included.

In Sections 2, 3 and 4, we will examine the volume rendering integral, and local and global illumination models used in volume rendering, respectively. In these sections, the discussions on lights and colors are wavelength-independent, and the optical models presented were often applied directly to RGB-based implementations in practice. In Section 5, we review the developments on spectral volume rendering, in which optical models were implemented in a wavelength-dependent manner.

Scientific Visualization: Advanced Concepts.
Editor: Hans Hagen; pp. 259–274
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Center for Informatics (Germany)

2 The Volume Rendering Integral

2.1 Basic Optical Model

The simplest physical model is that of a collection of small non-reflecting opaque particles, glowing with color c, which are the only light sources in the scene. The extinction coefficient τ represents the differential probability of a viewing ray hitting a particle, so that the probability of the ray hitting a particle along an infinitesimal length ds is $\tau\ ds$. Max [40] gives a derivation of $\tau = \pi r^2 N$ in terms of the radius r and number density N of the particles. In particular, τ is proportional to N. For particles of constant radius and material, τ is then also proportional to the mass density ρ of the particles, so that $\tau = \rho\kappa$, where κ depends on the material.

The transparency $T(s)$ represents the probability that the viewing ray travels a distance s away from the viewpoint without hitting any particles. Since the events "not hitting between distances 0 and s" and "not hitting between distances s and $s + ds$" are independent, their probabilities multiply, so

$$T(s+ds) = T(s)\big(1 - \tau(s)ds\big).$$

Thus

$$dT = T(s+ds) - T(s) = -T(s)\tau(s)ds,$$

so

$$dT/T = -\tau(s)ds.$$

Using the initial condition $T(0) = 1$,

$$\ln(T(s)) = -\int_0^s \tau(u)\,du,$$

and

$$T(s) = \exp\left(-\int_0^s \tau(u)\,du\right). \tag{1}$$

By the same independent event argument, the probability of not hitting a particle between distances 0 and s, and then hitting one between distances s and $s+ds$, is $T(s)\tau(s)ds$. In this case, the color will be $c(s)$, the color of the particles at position s. The values of τ and c at a given 3D position are usually specified by transfer functions of the scalar variable being visualized. The ray may pass through the complete volume, and hit an opaque background at distance D with probability $T(D)$, in which case it will have the background color B (which may depend on the point hit). The expected color E for the ray is gotten by averaging the colors for all these hitting events, weighted by their probabilities:

$$E = \int_0^D T(s)\tau(s)c(s)\,ds + T(D)B. \tag{2}$$

(For another derivation of this, see [40].) The product $\tau(s)c(s)$ can be thought of as a spatially varying color $C(s)$ which already has the effects of the particle density factored in. In the compositing literature, C is called a "premultiplied" color, and c is called a "non-premultiplied" color. In volume rendering it is also common for the transfer function to specify a premultiplied color, in which case

$$E = \int_0^D T(s)C(s)\,ds + T(D)B. \tag{3}$$

2.2 Integration by Discrete Sampling

The simplest way of estimating the integral in Equation (2) is to subdivide the interval from 0 to D into n segments of equal length $\Delta s = D/n$ and approximate the integral as a Riemann sum:

$$\int_0^D T(s)\tau(s)c(s)\,ds \cong \sum_{i=0}^{n-1} T(s_i)\tau(s_i)c(s_i)\Delta s,$$

where s_i is a sample in the i^{th} segment. If $s_i = i\Delta s$ is at the left hand end of the i^{th} segment,

$$\begin{aligned} T(s_i) &= \exp\left(-\int_0^{s_i} \tau(u)\,du\right) \\ &= \prod_{j=1}^{i} \exp\left(-\int_{s_{j-1}}^{s_j} \tau(u)\,du\right). \end{aligned}$$

If we assume $\tau(s)$ is constant on each segment (a further approximation),

$$T(s_i) \cong \prod_{j=1}^{i} \exp\Big(-\tau(s_j)\Delta s\Big).$$

Thus we can compute E by the following Iteration A:

```
E = 0.
T = 1.
for i = 0 to n-1 do
    E = E + T*tau[i]c[i]
    T = T*exp(-tau[i]*DeltaS)
E = E + T*B
```

Early ray termination stops this computation when T becomes zero, or small enough so that further terms have negligible effect. A similar iteration can be written for back to front accumulation of the contributions, which does not need the temporary variable T, but cannot do early ray termination.

A further approximation is to estimate exp(x) by the first two terms in its Taylor series:

$$\exp(x) = 1 + x + x^2/2 + \cdots$$

so that

$$\exp(-\tau(s)\Delta s) \cong 1 - \tau(s)\Delta s. \qquad (4)$$

This approximation is only good when the product $\tau(s)\Delta s$ is small; otherwise it may cause visible artifacts. If it is made, the steps in the above iteration can be achieved in simple compositing hardware. The data volume is sliced by a series of equally-spaced planes into textured polygons that can be scan converted and composited in hardware.

On a regular cubic or rectilinear grid, the data at s_i can be interpolated from the eight surrounding grid values using 3D texture mapping hardware. If only 2D texture mapping is available, the slices can be taken along a set of parallel coordinate planes in the grid, passing through the aligned data samples. It is more accurate to interpolate the scalar data and then apply the transfer functions for color and opacity, rather than sample the color and opacity at the grid and then interpolate them. This is easily done in hardware using dependent textures.

2.3 Integration along Ray Segments

Garrity [15] showed how to trace a ray through a tetrahedral grid, by determining through which face it exits a tetrahedron, and following a neighbor pointer to the next adjacent tetrahedron. Weiler *et al.* [56] have implemented this on a GPU.

Assume that the i^{th} segment is bounded by ray distances s_i and s_{i+1} and thus has length $\Delta s_i = s_{i+1} - s_i$, and that within its tetrahedron, the optical properties c_i and τ_i are constant. Then substituting Equation (1) into Equation (2), separating out the transparency up to s_i, and then integrating, the contribution E_i of the i^{th} segment to E is

$$\begin{aligned}
E_i &= \int_{s_i}^{s_{i+1}} \tau_i c_i T(s)\,ds \\
&= \int_{s_i}^{s_{i+1}} \tau_i c_i \exp\left(-\int_0^s \tau(u)\,du\right) ds \\
&= \tau_i c_i \int_{s_i}^{s_{i+1}} \exp\left(-\int_0^{s_i} \tau(u)\,du\right)\cdot\exp\left(-\int_{s_i}^{s} \tau_i\,du\right) ds \\
&= T(s_i)\tau_i c_i \int_{s_i}^{s_{i+1}} \exp(-\tau_i(s-s_i)\,ds \\
&= T(s_i)\tau_i c_i \left.\frac{\exp(-\tau_i(s-s_i))}{-\tau_i}\right|_{s_i}^{s_{i+1}} \\
&= T(s_i)c_i\big(1-\exp(-\tau_i\Delta s_i)\big).
\end{aligned}$$

The following Iteration B can then compute the radiance along the ray with floating point accuracy:

```
E = 0.
T = 1.
for i = 0 to n-1 do
    F = exp(-tau[i]*DeltaS[i])
    E = E + T*c[i]*(1.  - F)
    T = T*F
E = E + T*B
```

Even for the discrete sampling in the previous section, Iteration B will be more accurate than the Iteration A derived from the Riemann sum, if $\tau\Delta s$ is large, so that T varies significantly between samples.

Shirley and Tuchman [52] divided the projection of a tetrahedron by its projected edges into triangles in software, and used polygon rendering and compositing to render them in hardware. Wylie *et al.* [60] have now done all steps in hardware, except the global visibility sorting required to determine the compositing order. The same principles apply to a mesh of arbitrary convex polyhedral cells, as shown by Williams *et al.* [59], which also presents a visibility sort based on a method proposed by Martin Newell.

This visibility sorting remains a bottleneck, and in fact is not always possible, because there may be a cycle of tetrahedra, each overlapping the next. Visibility sorting has its own extensive literature, which is beyond the scope of this survey.

2.4 Analytic Integration

So far we have assumed that the optical properties are constant within each cell. Another case which can be handled analytically is when they vary linearly along each ray. For example, this is the case if the scalar function is linearly interpolated from its values at the four vertices of a tetrahedron, and the transfer functions mapping it to τ and c or C are also linear. In such cases, applying a symbolic integration package to the integral in Equation (2) produces a rather complicated analytic expression, first presented by Williams and Max [58]. As shown in [59], this method also applies to piecewise linear transfer functions, by dividing the tetrahedra into sub-polyhedra in which the transfer function is linear. That paper also improves the numerical stability of the software evaluation. Moreland and Angel [43] have used a GPU to evaluate this analytic expression in hardware, by precomputing part of it into texture tables.

2.5 Other Integration Methods

If the volume data is determined at grid points, the discrete sampling methods in Section 2.2 are appropriate, but if the data comes from a finite element grid, integration along the ray segments in each grid cell may be more accurate and efficient. However, only the constant case in Section 2.3 and the linear case in Section 2.4 admit analytic solutions. The transfer function may not be linear or even piecewise linear. Also, the interpolated scalar value along a ray may not be linear. For example, for trilinear interpolation on hexahedral elements, the scalar function is a cubic polynomial along general rays. In addition, higher order finite elements may be used in the physical simulation for better convergence.

A further problem with finite elements is that they may be deformed, that is, the mapping from a standard element shape like a cube or regular tetrahedron into the physical simulation space may be non-linear. Often, the same higher order polynomials used to define the scalar function on the standard element are also used to map the standard element into physical space. Thus, an inverse map is needed before the scalar function can even be evaluated at a point on the ray. In fact, a ray can intersect a deformed element in more than one segment. Wiley *et al.* [57] show how to determine these segments for the case of deformed quadratic tetrahedra, and approximate the inverse mapping along them by polynomial splines.

Once the ray segments are known, there are several ways that the integral can be approximated. In [57], the equally spaced sampling method of Section 2.2 was used. But for smoothly varying transfer functions, Gaussian integration [48] can give greater accuracy for the same number of sample points, by approximating the integral as an unequally weighted sum of the integrand at n unequally spaced sample points. The sample locations and sample weights are chosen to give the exact integral for polynomials of the maximum degree, $2n-1$, allowed by the $2n$ degrees of freedom in these positions and weights. So if the integrand is well approximated by such polynomials, Gaussian integration will give a good approximation. But this is not always the case, especially for non-smooth transfer functions that are selected to emphasize certain contour surfaces of the scalar function. For this application, discussed in more detail in the next section, it may be more appropriate to find the exact intersection of the ray with any contour surface within the volume element. Williams *et al.* [59] used the quadratic formula to intersect rays with contour surfaces within undeformed tetrahedra with quadratically varying scalar functions. Kirby and Nelson [28] estimate the scaler function along the inverse-mapped ray within a higher order deformed element by a high degree polynomial, and then use a general root finding procedure to find its first root along the ray.

Another possibility is to precompute and store the integrals. For a linear variation of the scalar function along a ray segment and arbitrary transfer functions, the integral along the segment depends only on the segment length, and the scalar values at its two endpoints. Röttger *et al.* [50] propose storing the integrals in a 3D texture, indexed by these three variables. Weiler *et al.* [56] used this in their hardware ray tracing, and proposed an efficient incremental method of doing the precomputation. For the hardware implementation by plane slicing given at the end of Section 2.2, the ray segment length is constant in an orthogonal view, and approximately constant in a perspective view, so only a 2D texture is required for the precomputed integrals. A good reference on hardware implementation of the methods in this and the next two sections is [13].

3 Local Illumination

The volume rendering integrals given in Equations (2) and (3) represent a typical optical model which is referred to as the *absorption plus emission model* [40] and is perhaps the most widely-used optical model in volume visualization. The non-premultiplied color $c(s)$ can simply be a function of emission property at a sample point s by considering the volume as a light-emitting medium. Another approach is to involve one or more light sources in the computation of $c(s)$. In both cases, the illumination at s depends not only the optical properties sampled at s and the intensity of each light source, but also indirect light reflected towards s from other part of the medium (i.e., scattering) as well as the absorptivity of the medium that determines how much light can eventually arrived at s (i.e., shadows). Such an illumination model is referred to as *global illumination*, which will be discussed in detail in Section 4.

To avoid costly computation with a global illumination model, it is common to adopt a *local illumination model* where $c(s)$ is estimated based only on the optical properties sampled at s and the intensity of each light source. This allows us to rewrite Equation (2) as:

$$E = \int_0^D T(s)\tau(s)\left(c_g(s) + \sum_{i=1}^{k} c_r(s, L_i)\right) ds + T(D)B,$$

where $c_g(s)$ is the sampled intensity of self-glowing at s, and $c_r(s, L_i)$ defines the light reflected due to light source $L_i\,(i = 1, \ldots, k)$. In many applications, a local illumination model is normally adequate for rendering a single isosurface within a volume. When handling multiple isosurfaces, or amorphous regions, one needs to aware the limitation of such a model and the potential perceptual discrepancy due to the omission of shadows and indirect lighting.

3.1 Classic Illumination Models

Given a light source L, one can estimate c_r at s locally by using one of the empirical or physically-based illumination models designed for surface geometry, such as the Phong, Phong-Blinn, and Cook-Torrance models [14]. When such a model is used in volume rendering, it is assumed that each sampling position, s, is associated with a surface or microfacet. This assumption allows us to compute the surface normal at s, which is required by almost all surface-based illumination models.

In volume models, surface geometry is normally not explicitly defined, and in many situations, models do not even assume the existence of a surface. Hence, the computation of surface normals is usually substituted by that of gradient vectors. While for some parametric or procedurally-defined volume models, it is possible to derive gradient vectors analytically, in most applications, especially where discrete volumetric models are used, gradient vectors are estimated, for example, using the finite differences method for rectangular grids [41], and 4D linear regression for both regular and irregular grids [44]. The commonly used central differences method is a reduced form of finite differences based on the first two terms of the Taylor series. There are many other gradient estimation methods, including schemes that involve more or fewer neighboring samples (e.g., [47, 63]), and schemes where the discrete volume models are first convolved using a high-order interpolation function, and gradients are computed as the first derivative of the interpolation function (e.g., [6, 45]). Möller *et al.* compared a few normal estimation schemes in the context of volume visualization [42].

Usually the local illumination models are only applied at or near a presumed surface within the volume, so a surface presence indicator is used to weight (i.e. multiply) the computed local illumination. Levoy [36] describes two methods of computing this surface weight, using formulas involving the scalar value and its gradient magnitude, and Kindleman and Durkin [27] give a method which uses a 2D texture table indexed by these two quantities. Kraus [32] points out that if the task is to determine only if a contour surfaces is intersected by a ray segment, this can be indicated by a

2D texture table indexed by the two scalar values at the endpoints of the segment. Alternatively, ray tracing can be used to locate points exactly on an isosurface, as described in Section 2.5, and the local illumination can be applied only at these points.

3.2 Measured and Precomputed BRDFs

The light reflected from a point on a surface can be described by a *bi-directional reflection distribution function* (BRDF). Hence, it is feasible to obtain a BRDF in sampled form by either measurement or computer simulation [16]. The measurements of a BRDF are usually made using a goniophotometer in a large number of directions, in terms of polar and azimuth angles, uniformly distributed on a hemisphere about a source [20]. In computer graphics, it is also common to precompute discrete samples of a BRDF on a hemisphere surrounding a surface element (e.g., [25, 4]).

Given n sampling points on a hemisphere, and n possible incident directions of light, a BRDF can be represented by an $n \times n$ matrix. Given an arbitrary incident light vector, and an arbitrary viewing vector, one can determine the local illuminance along the viewing vector by performing two look-up operations and interpolating up to 16 samples.

One major advantage of using measured or pre-computed BRDFs is that the computation of $c(s)$ in Equation (2) or $C(s)$ in Equation (3) no longer needs to rely on an illumination model that can easily be defined and computed. One can use measured data to compensate for the lack of an appropriate model that accounts for a range of physical attributes, or use precomputed data for a complicated and computationally intensive model (e.g., an anisotropic model as in [25]).

Similar to a BRDF, the light transmitted at a point on a surface can be described by a *bi-directional transmittance distribution function* (BTDF). Hanrahan and Krueger [17] considered both BRDF and BTDF in a multi-layered surface model, which can be viewed as a simplified volume model. Baranoski and Rokne [3] applied this approach to the modeling of foliar scattering. Wang *et al.* [55] obtained their BRDFs and BTDFs by fitting parametric models to measured reflectance and transmittance data.

3.3 Phase Functions

A phase function, $r(\omega, \omega')$, defines a probability distribution of scattering in direction ω with respect to the direction of the incident light, ω' [10]. More precisely, $r(\omega, \omega')d\omega$ represents the probability that light flowing in direction ω' that scatters from a particle scatters into the solid angle $d\omega$ about the direction ω. Blinn [10] gives the phase function for a spherical diffusely reflecting particle. The Henyey-Greenstein phase function [19] is also popular in computer graphics. A discussion of the phase function in the context of multiple scattering will be given in Section 4.1. Here we briefly describe its use as a local illumination model.

The fundamental difference between such an illumination model and those in 3.1 and 3.2 is that it is entirely volumetric and does not assume the existence of a surface or microfacet at every visible point in space. While phase functions are largely used in the context of global illumination, they can be used as for local illumination in a perhaps rather simplified manner. Despite the omission of the multiple scattering feature in the context of local illumination, phase functions allow a volumetric point to be lit by light from any direction. On the contrary, classic illumination models and BRDFs consider only light in front of the assumed surface or microfacet defined at the point concerned.

3.4 Other Related Developments

Many local illumination methods developed for surface rendering have been, or can be, used in conjunction with the volume rendering integral. These include Blinn and Newell's environment map [9], Arvo's illumination map [2], Heckbert's radiosity texture [18]. However, the application of these methods in volume rendering has so far been largely limited to the rendering of a single iso-surface [31].

The discrete sampling process described in Section 2.2 facilitates a scattering event at each sample. In such a process, refraction can be realized as a local illumination feature by altering the ray path. Rodgman and Chen examined several sampling methods for rendering refraction in conjunction with the volume rendering integral [49], and employed nonlinear diffusion filters to improve the quality of refraction rendering. Li and Mueller studied the use of different interpolation filters in a surface-based approach to rendering refraction [38, 37].

In many visualization applications, it is appropriate, and often desirable, to use *non-photorealistic lighting* to enhance the perception of synthesized visualizations. Recently, Stewart employed local occuluders and uniform diffuse illumination to render pseudo-shadows in depressions and crevices [54]. Lee *et al.* used globally inconsistent lighting to enhance perception of shapes [35]. Lum *et al.* applied transfer functions to incoming light to provide better perception of material thickness and boundaries [39].

4 Global Illumination

4.1 The Basic Equation for Multiple Scattering

If the particles in the optical model scatter as well as emit light, the mathematical situation is more complicated than in section 2. For a complete solution, we must determine $I(x,\omega)$, the radiance (light intensity) flowing through every 3D point x in the volume, in every direction ω on the unit sphere, taking into account the effects of multiple scattering.

The probability that light hitting a particle scatters instead of being absorbed is called the albedo a. The scattering depends on the direction of the incoming and scattering rays according to the phase function $r(\omega,\omega')$.

Let the source function $g(x,\omega)$ represent the light emitted or inscattered into direction ω by a particle at position x. Integrating the scattered incoming light over all incoming directions ω' in the unit sphere Ω, taking into account the effects of the albedo and the phase function, and finally adding on the glow $c_g(x)$, we get

$$g(x,\omega) = \int_{\Omega} a(x)r(\omega,\omega')I(x,\omega')\,d\omega' + c_g(x). \tag{5}$$

In practice, a usually does not depend on x, and the phase function $r(\omega,\omega')$ depends only on the scattering angle between the unit direction vectors ω and ω'.

In order to write the multiple scattering version of Equation (2), giving an expression for $I(x,\omega)$, we substitute $g(x,\omega)$ for $c(s)$. We evaluate the optical properties at points $x(s) = x - \omega s$, since the integral in (2) is along a “viewing ray” in the direction opposite to the light flow. Thus we have

$$I(x,\omega) = \int_{0}^{D} T(s)\tau(x(s))g(x(s),\omega)\,ds + T(D)B. \tag{6}$$

This equation is difficult to solve, since $g(x(s),\omega)$ depends via Equation (5) on $I(x,\omega')$ at all the points $x(s)$ on the viewing ray and in all directions ω' in the unit sphere, so that all the $I(x,\omega)$ must be solved for simultaneously. Surveys of techniques for the solution are given in Pérez *et al.* [46] and Max [40].

4.2 Single Scattering Approximation

One simplifying assumption is that the albedo and/or density is low, so that the probability of light scattering more than once is small, and only single scattering need be considered. For further simplicity here, we will also assume that there is no glow c_g, and only one parallel light source with intensity L_0 at an infinite distance in the single direction ω'. Then we can precompute the shadowing effects of the volume opacity, and determine the $I(x,\omega')$ inside the integral in Equation (5) by using Equation (1) to obtain

$$I(x,\omega') = L_0 \exp\left(-\int_0^D \tau(x - s\omega')\, ds\right).$$

Kajiya and von Herzen [26] did this in a first pass through the volume, essentially computing $I(x,\omega')$ by the part of Iteration A involving only T. Bahrens and Rattering [5] give an accurate and efficient method for this shadow pre-computation on gridded data, using texture mapping hardware, by moving a sampling plane perpendicular to ω' in discrete steps along the light flow direction. This shadow computation can be done in the same pass as the volume rendering if the sampling plane instead bisects the angle between the viewing and illumination rays, as in [29].

4.3 Diffusion Approximation

Another simplifying assumption is that the albedo is high, and that the size of the volume features is large compared to the mean free path $1/\tau$. Then almost every ray seen at the viewpoint will have been scattered so many times that all the directional properties of $r(\omega,\omega')$ will be washed out by multiple spherical convolutions, and the scattering will be effectively diffuse, equal in all directions. In this case, the flow of light can be modeled by a second order partial differential equation for $I(x)$ as a function of position only. Stam [53] first introduced this equation to computer graphics, and solved it using the multi-grid method. (His equation is off by a factor of 3 in a couple of the terms. For a correct derivation see Ishimaru [21].)

Jensen *et al.* [24] introduced an approximate solution to this equation based on fitting the analytic solution for the diffusion from two virtual point light sources, one below a planar surface bounding the participating medium, and one above it, outside the medium. This approximation is only valid for a semi-infinite domain with constant optical properties, bounded by a flat plane, but it has been applied to give realistic renderings for other geometries. Jensen and Buhler [22] used an octree hierarchy to account for illumination at all the surface points, and this idea has now been refined by various authors, for example Dachsbacher and Stamminger [11], to give real time performance on graphics hardware. However, the basic technique of Jensen *et al.* [24] is only applicable to homogenous materials, and is less useful for volume rendering of spatially varying data.

4.4 Other Multiple Scattering Methods

For non-homogeneous materials, there are several methods not covered in the survey of Perez *et al.* [46]. Jensen and Christiansen [23] extend photon mapping to participating media. They do a Monte Carlo simulation of photon transport from the light sources, which can take into account spatially varying optical properties and general phase functions. They record each scattering event in a spatial data structure called a photon map. In a final gather pass from the viewpoint, they collect the scattering events relevant to a viewing ray, to account for the inscattered photons. Of course, this can be slow, since many photons are required for accurate convergence.

Kniss *et al.* [30] have generalized the single pass shadow algorithms given at the end of Section 4.2 to multiple forward scattering, by gathering the accumulated light from several sampled directions in the previous sampling plane. Zhang *et al.* [62] have added backward scattering from the next couple of sampling planes.

5 Spectral Volume Rendering

We now consider visible light as a form of electromagnetic radiation. The radiative power emitted by an object is typically defined as a function known as the *spectral power distribution (SPD)*, $F(\lambda)$, where λ is the wavelength within the radiation band concerned. In color computation, it is common to limit this range to the visible spectrum, $\lambda \in [380nm, 770nm]$, or often a narrower range, $\lambda \in [400nm, 700nm]$, to which human eyes are more sensitive.

5.1 Basic Optical Models

The transmission of light of a single wavelength λ through a homogeneous isotropic absorption filter (such as glass and gelatin) is governed by the Bouguer's or Lambert's law (1727, 1760), which states that the intensity of an incoming light, $L_0(\lambda)$ decreases exponentially with the path length s in the filter medium, that is:

$$L(\lambda) = L_0(\lambda) \cdot \exp\big(-s \cdot \tau(\lambda)\big),$$

where $\tau(\lambda)$ is the spectral extinction coefficient (commonly referred to as absorptivity) of the medium. Beer (1983) extended the Lambert-Bouguer law to a liquid solution with a low or moderate concentration of an absorbing solute, as:

$$L(\lambda) = L_0(\lambda) \cdot \exp\big(-s \cdot \nu \cdot \kappa(\lambda)\big),$$

where ν represents the concentration of the solute and $\kappa(\lambda)$ is the extinction coefficient of the solute [61].

When the transparency $T(\lambda, \Delta) = \exp\big(-\Delta \cdot \tau(\lambda)\big)$ is known for a standard path length Δ, we can obtain the transparency for an arbitrary path length δ as:

$$T(\lambda, \delta) = T(\lambda, \Delta)^{\frac{\delta}{\Delta}}.$$

This is more often written in the form of a depth correction formula for opacity, $\alpha(\lambda, \delta)$, as:

$$\alpha(\lambda, \delta) = 1 - T(\lambda, \delta) = 1 - \big(1 - \alpha(\lambda, \Delta)\big)^{\frac{\delta}{\Delta}}.$$

For a homogeneous medium, the Lambert-Bouguer law, which was derived from experimentation, is consistent with Equation (1), which was derived independently based on the notion of absorbing particles. Comparing with the early discussion in Section 2.1, the Lambert-Bouguer law is a special case of Equation (1), while transparency $T(s)$ in Equation (1) can be considered as an approximated extension of the Lambert-Bouguer law by removing the homogeneity condition and assuming the same refractive index for materials with different $\tau(s)$.

The basic optical model proposed by Bouguer and Lambert, and a wavelength-dependent version of Equation (1) form the basis of the two spectral volume rendering integrals in 5.2 and 5.3.

5.2 A One-pass Rendering Integral

Bergner *et al.* [7, 8] developed a spectral volume rendering integral, partially based on the multiple scattering model described in Section 4.1. In order to facilitate ray casting with local illumination, they simplified the multiple scattering model by removing all the global illumination elements in the model. This includes (*i*) assigning volumetric shadow ratio to constant 1 for all voxels and directions, (*ii*) considering only irradiance and radiance with the same direction as the viewing ray, and (*iii*) approximating the radiance at each point by the reflectance of the local materials.

In particular, Bergner *et al.* successfully factored the light source out in the computation of their volume rendering integral. This enables interactive data integration using different light sources during post illumination.

Let $L(\lambda)$ be a light source of a single wavelength, and $I_i(x,\omega,\lambda)$ be the irradiance at a sampling position x reached from a light ray of direction ω. I_i is computed as:

$$I_i(x,\omega,\lambda) = L(\lambda)\cdot\tilde{I}_i(x,\omega,\lambda,D),$$

where D is the distance between x and the point d where the light ray first enters the volume. $\tilde{I}_i(x,\omega,\lambda)$ indicates the proportion of $L(\lambda)$ that has arrived at x from d, which is:

$$\tilde{I}_i(x,\omega,\lambda) = \tilde{I}_i(x-D\omega,\omega,\lambda)\cdot T(x,\omega,\lambda,D) + \int_0^D \tilde{I}_r(x-s\omega,\omega,\lambda)\cdot T(x,\omega,\lambda,s)\,ds,$$

where $\tilde{I}_i(x-D\omega,\omega,\lambda)$ is the irradiance at d, $T(x,\omega,\lambda,s)$ is the spectral version of the extinction function in Equation (1), i.e.,

$$T(x,\omega,\lambda,s) = \exp\left(-\int_0^s \tau(x-t\omega,\lambda)dt\right),$$

and $\tilde{I}_r(x-s\omega,\omega,\lambda)$ indicates the scattered radiance from the external illumination. Because of the abovementioned simplification, the term $\tilde{I}_r(x-s\omega,\omega,\lambda)$ is computed with a local illumination model based on the Phong model.

5.3 A Two-pass Spectral Rendering Integral

Noordmans *et al.* considered a spectral volume rendering integral which features shadow computation. They adopted the Kajiya and von Herzen's two pass algorithm [26] for computing volumetric shadows and adapted the notion of opacity in the traditional RGBα-based volume rendering. Let $\rho_m(x)$ be the mass density of a material m at x, and $\kappa_m(\lambda)$ be the absorption attribute of m at wavelength λ. Consider the position x features M materials. The spectral representation of opacity at x is defined as:

$$\tau(x,\lambda) = \sum_{m=1}^{M} \rho_m(x)\cdot\kappa_m(\lambda).$$

In the *illumination* phase, the flux of the light is propagated through the volume using discrete ray casting. At each sampling position x, the incident light I_i is related to the transmitted light I_t (which subsequently becomes the incident light at the next sample) simply as:

$$I_i(x+\Delta x,\lambda) = I_t(x,\lambda) = I_i\cdot\big(1-\Delta x\cdot\tau(x,\lambda)\big).$$

In the *radiation* phase, the flux of radiance is accumulated also using back-to-front ray casting. At each sampling position x, the radiance I_r is related to the irradiance I_b arriving from the previous sample, and the local emission I_e as:

$$I_b(x+\Delta x,\lambda) = I_r(x,\lambda) = I_e(x,\lambda) + I_b(x,\lambda)\cdot\big(1-\Delta x\cdot\tau(x,\lambda)\big).$$

Note that as mentioned in conjunction with Equation (4), the term $1-\Delta x\cdot\tau(x,\lambda)$ used in both phases is only an approximation.

One particular interesting aspect of the work by Noordmans *et al.* is the design of its emission function, which is split into two parts, namely elastic scattering and inelastic scattering. The former features scattering in a chromatic medium, with each material m is associated with a specific spectral band, facilitating a spectral transfer function. The latter enables materials to absorb the incident light at one wavelength and re-emit the energy at another, facilitating a simulation of fluorescence and phosphorescence materials.

5.4 One-dimensional Radiosity

Abdul-Rahman and Chen [1] presented a spectral volume rendering integral based on the optical model proposed by Kubelka and Munk [34], commonly referred to as the Kubelka and Munk theory. The theory, which was built upon Schuster's two flux concept [51], differs from the Lambert-Bouguer law in at least two respects. (i) It models both absorption and scattering but only in the directions of an *incident flux* and a *reflected flux*. (ii) It assumes that a volumetric colorant layer can be divided into a large number of homogeneous elementary layers, the optical properties of the volume thus depend on one direction, say x. The two fluxes I_i and I_r flow in opposite directions, x and $-x$ respectively.

Consider an isotropic elementary layer of thickness dx, which is associated with an absorbing coefficient $K(\lambda)$ and a scattering coefficient $S(\lambda)$. With the incident flux $I_i(\lambda)$, the passage of a light beam through the layer will have its energy decreased through absorption, by an amount of $K(\lambda)I_i(\lambda)\,dx$, and through scattering, by an amount of $S(\lambda)I_i(\lambda)\,dx$. At the same time, because of the radiation reflected by the reflected flux $I_r(\lambda)$, the energy is also increased due to backscatter, by an amount of $S(\lambda)I_r(\lambda)\,dx$. This gives the total changes of the $I_i(\lambda)$ as:

$$dI_i(\lambda) = -K(\lambda)I_i(\lambda)\,dx - S(\lambda)I_i(\lambda)\,dx + S(\lambda)I_r(\lambda)\,dx. \tag{7}$$

On the other hand, the passage of the reflected flux, $I_r(\lambda)$, in the opposite direction, is also subject to similar changes, that is:

$$-dI_r(\lambda) = -K(\lambda)I_r(\lambda)\,dx - S(\lambda)I_r(\lambda)\,dx + S(\lambda)I_i(\lambda)\,dx. \tag{8}$$

Note that when $S(\lambda) = 0$, the incident flux, on integrating, follows the Lambert-Bouguer law discussed in 5.1. Also note that without the back-scattering, we would have $K(\lambda) + S(\lambda) = \tau(\lambda)$, where $\tau(\lambda)$ is the extinction coefficient in the Lambert-Bouguer law. As placement of the colorant layer and its elementary layers are normally drawn in a horizontal manner, the incident and reflected fluxes are commonly referred to as downward and upward fluxes respectively.

Dividing both sides of Equations (7) and (8) by dx, we have two simultaneous differential equations. The solution to the equations leads to several useful formulae. Consider a single homogeneous layer of thickness X. Let $R(\lambda)$ and $T(\lambda)$ be the reflectance and transmittance of the layer respectively. We have:

$$R(\lambda) = \frac{\sinh(b(\lambda)S(\lambda)X)}{a(\lambda)\sinh(b(\lambda)S(\lambda)X) + b(\lambda)\cosh(b(\lambda)S(\lambda)X)} \tag{9}$$

$$T(\lambda) = \frac{b(\lambda)}{a(\lambda)\sinh(b(\lambda)S(\lambda)X) + b(\lambda)\cosh(b(\lambda)S(\lambda)X)}, \tag{10}$$

where

$$a(\lambda) = \frac{S(\lambda) + K(\lambda)}{S(\lambda)}, \quad b(\lambda) = \sqrt{a(\lambda)^2 - 1}.$$

Here $a(\lambda)$ is essentially a spectral version of the albedo defined in 4.1, but only the probability of hitting back-scatters is considered. The reflectance of an opaque medium can thereby be computed by making $X \to \infty$ in Equation (9), resulting in:

$$R_\infty(\lambda) = 1 + \left(\frac{K(\lambda)}{S(\lambda)}\right) - \left[\left(\frac{K(\lambda)}{S(\lambda)}\right)^2 + 2\left(\frac{K(\lambda)}{S(\lambda)}\right)\right]^{\frac{1}{2}}.$$

Kubelka later extended the Kubelka-Munk theory to inhomogeneous layers [33]. Given the reflectance and transmittance of two different layers, R_1, T_1, R_2 and T_2, considering an infinite process of interaction between the two layers. A light flux passes the first layer with the portion T_1, and then the second layer T_1T_2. Meanwhile, a portion of $T_1R_2T_1$ will be reflected from the second layer and pass though the first layer again. Continuing this process leads to two infinite series:

$$\begin{array}{cccc} R_1, & T_1R_2T_1, & T_1R_2R_1R_2T_1, & \dots \\ T_1T_2, & T_1R_2R_1T_2, & T_1R_2R_1R_2R_1T_2, & \dots \end{array}$$

The infinite process of interaction can be considered as one-dimensional radiosity [12] — a limited form of global illumination. The sums of the two series give the compositing reflectance and transmittance as:

$$R(\lambda) = R_1(\lambda) + \frac{T_1(\lambda)^2R_2(\lambda)}{1-R_1(\lambda)R_2(\lambda)}$$

$$T(\lambda) = \frac{T_1(\lambda)T_2(\lambda)}{1-R_1(\lambda)R_2(\lambda)}.$$

A discrete spectral volume rendering integral can thus be formulated based on this multi-layer model. Using the ray-casting method, one can approximate the intersection volume between each ray and a volume object as a series of homogeneous layers with a thickness equal to the sampling distance.

The work by Abdul-Rahman and Chen [1] highlighted some relative merits of the Kubelka-Munk theory over the Lambert-Bouguer law, especially in terms of its built-in distance attenuation, and its capability of determining the opacity and transparency optically according to the absorption and scattering properties. They have also experimented with captured optical properties of some real world materials, and the design for spectral transfer function for post-illumination.

6 Summary and Conclusions

Volume rendering is an indispensable tool for synthesizing images involved volumetric models. Much of the theoretic foundation was laid down in late 1980's and early 1990's [40]. Until recent years, however, most volume rendering systems employed only basic local illumination models. Due to the rapid increase of computation power, a collection of complex illumination features, such as shadows and refraction, have started to appear in some recent developments. This survey has provided an overview of optical and illumination models for volume rendering, while highlighting some new developments including spectral volume rendering, shadow algorithms and refraction rendering. We hope that this survey will encourage further research into the development and use of complex illumination models to achieve better realism and perception through optical correctness.

Acknowledgments

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract number W-7405-ENG-48. It was also partly funded by a UK-EPSRC grant with reference GR/S44198. The authors wish to thank Dr. Martin Kraus, Universität Stuttgart, for his comments and suggestions in the early stage of preparing this survey.

References

1 A. Abdul-Rahman and M. Chen. Spectral volume rendering based on the Kubelka-Munk theory. *Computer Graphics Forum*, 24(3), 2005.

2 James Arvo. Backward ray tracing. In *SIGGRAPH '86 Developments in Ray Tracing, Seminar Notes*, August 1986.

3 G. V. G. Baranoski and J. G. Rokne. An algorithmic reflectance and transmittance model for plant tissue. *Computer Graphics Forum*, 16(3):141–150, 1997.

4 B. G. Becker and N. L. Max. Smooth transitions between bump rendering algorithms. *ACM SIGGRAPH Computer Graphics*, 27(3):193–190, 1993.

5 Uwe Behrens and Ralf Ratering. Adding shadows to a texture-based volume renderer. In *Proceedings ACM SIGGRAPH Symposium on Volume Visualization*, pages 39–46, 1998.

6 M. J. Bentum, B. B. A. Lichtenbelt, and T. Malzbender. Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, 1996.

7 S. Bergner, T. Möller, M. S. Drew, and G. D. Finlayson. Internative spectral volume rendering. In *Proc. IEEE Visualization*, pages 101–108, 2002.

8 S. Bergner, T. Möller, M. Tory, and M.S. Drew. A practical approach to spectral volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):207–216, 2005.

9 James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976.

10 J.F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. *Computer Graphics*, 16(3):21–29, 1982.

11 Carsten Dachsbacher and Marc Stamminger. Translucent shadow maps. In *Rendering Techniques 2003*, pages 197–201. Eurographics, 2003.

12 J. Dorsey and P. Hanrahan. Modeling and rendering of metallic patinas. In *Proc. 23rd Annual ACM SIGGRAPH Conference*, pages 387–396, 1996.

13 Klaus Engel, Markus Hadwiger, Joe Kniss, Christof Rezk-Salama, and Daniel Weiskopf. *Real-Time Volume Graphics*. A. K. Peters, 2006.

14 J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 2nd edition, 1990.

15 Michael P. Garrity. Ray tracing irregular volume data. *ACM SIGGRAPH Computer Graphics*, 24(5):35–40, 1990.

16 D. P. Greenberg, J. Arvo, E. Lafortune, K. E. Torrance, J. A. Ferwerda, B. Walter, B. Trumbore, P. Shieley, S. Pattanaik, and S. Foo. A framework for realistic image synthesis. In *Proc. ACM SIGGRAPH Annual Conference Series*, pages 477–494, 1997.

17 Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of ACM SIGGRAPH Annual Conference*, pages 165–174, 1993.

18 Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics (SIGGRAPH '90)*, 24(4):145–154, 1990.

19 G. L. Henyey and J. L. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, 88:70–73, 1940.

20 R. S. Hunter and R. W. Harold. *The Measurement of Appearance*. John Wiley & Sons, 2nd edition, 1987.

21 Akira Ishimaru. *Wave Propagation and Scattering in Random Media, Volume I: Single Scattering and Transport Theory*. Academic Press, New York, 1978.

22 Henrik Wann Jensen and Juan Buhler. A rapid hierarchical rendering technique for translucent materials. *ACM Transactions on Graphics (SIGGRAPH '2002)*, 21(3):721–728, 2002.

23 Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of ACM SIGGRAPH Annual Conference*, pages 311–320, 1998.

24 Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH Annual Conference*, pages 511–518, 2001.

25 J. T. Kajiya. Anisotropic reflection models. *ACM SIGGRAPH Computer Graphics*, 19(3):15–21, 1985.

26 J. T. Kajiya and B. P. von Herzen. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, 18(3):165–174, 1984.

27 Gordon Kindleman and James Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings, IEEE/ACM Symposium on Volume Visualization*, pages 79–86, 1998.

28 Robert M. Kirby and Blake Nelson. Ray-tracing polymorphic multi-domain spectral/hp elements for isosurface rendering. *IEEE Transcations on Visualization and Computer Graphics*, 12(1), 2006.

29 Joe Kniss, Gordon Kindlmann, and Charles Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

30 Joe Kniss, Simon Premoze, Charles Hansen, and David Ebert. Interactive translucent volume rendering and procedural modeling. In *Proceedings, IEEE Visualization*, pages 109–116, Washington, DC, USA, 2002. IEEE Computer Society.

31 Joe Kniss, Christof Rezk-Salama, Klaus Engel, and Markus Hadwiger. High-quality volume graphics on consumer PC hardware. In *SIGGRAPH '2002 Course*, 2002.

32 Martin Kraus. Scale-invariant volume rendering. In *Proceedings, IEEE Visualization*, pages 295–302, 2005.

33 P. Kubelka. New contributions to the optics of intensely light-scattering materials, Part II. nonhomogeneous layers. *Journal of the Optical Society of America*, 44:330 – 355, 1954.

34 P. Kubelka and F. Munk. Ein Beitrag zur Optik der Farbanstriche. *Zeitschrift für Technische Physik*, 12:593–601, 1931.

35 Chang Ha Lee, Xuejun Hao, and Amitabh Varshney. Light collages: Lighting design for effective visualization. In *Proc. IEEE Visualization*, pages 281–288, 2004.

36 M. Levoy. Volume rendering: display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

37 Shengying Li and Klaus Mueller. Accelerated, high-quality refraction computations for volume graphics. In *Proc. Volume Graphics*, pages 73–81. Eurographics, 2005.

38 Shengying Li and Klaus Mueller. Spline-based gradient filters for high-quality refraction computation in discrete datasets. In *Proc. EuroVis*, pages 215–222. Eurographics, 2005.

39 Eric B. Lum and Kwan-Liu Ma. Lighting transfer functions using gradient aligned sampling. In *Proc. IEEE Visualization*, pages 289–296, 2004.

40 N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

41 T. H. Meyer, M. Eriksson, and R. C. Maggio. Gradient estimation from irregularly spaced data sets. *Mathematical Geology*, 33(6):693–717, 2001.

42 Torsten Möller, Raghu Machiraju, Klaus Mueller, and Roni Yagel. A comparison of normal estimation schemes. In *Proc. IEEE Visualization '97*, pages 19–26, Phoenix, AZ, November 1997.

43 Kenneth Moreland and Edward Angel. A fast high accuracy volume renderer for unstructured data. In *Proceedings of the IEEE/SIGGRAPH Symposium on Volume Visualization*, pages 9–16, 2004.

44 L. Neumann, B. Csébfalvi, A. König, and E. Gröller. Gradient estimation in volume data using 4D linear regression. *Computer Graphics Forum*, 19(3):C351–C357, 2000.

45 S. K. Park and R. A. Schowengerdt. Image reconstruction by parametric cubic convolution. *Computer Vision, Graphics and Image Processing*, 23:258–272, 1983.

46 Frederic Pérez, Xaview Pueyo, and François Sillion. Global illumination techniques for the simulation of participating media. In Julie Dorsey and Philipp Slusallek, editors, *Rendering Techniques '97*, pages 309–320, Vienna, 1997. Springer.

47 A. Pommert, U. Tiede, G. Wiebecke, and K. H. Hohne. Surface shading in tomographic volume visualization. In *Proc. 1st Conference on Visualization in Biomedical Computing*, volume 1, pages 19–26, 1990.

48 William Press, Saul Teukolosky, William Vettering, and Brian Flannery. *Numerical Recipes in C++*. Cambridge University Press, 2002.

49 David Rodgman and Min Chen. Refraction in discrete raytracing. In Klaus Mueller and Arie Kaufman, editors, *Proc. Volume Graphics 2001*, pages 3–17. Springer, 2001.

50 Stefan Röttger, Martin Kraus, and Thomas Ertl. Hardware-accelerated volume and isosurfaces rendering based on cell projection. In *Proc. IEEE Visualization*, pages 109–116, 2000.

51 A. Schuster. Radiation through a foggy atmosphere. *Journal of Astrophysics*, 21:1–22, 1905.

52 Peter Shirley and Alan Tuchman. A polygonal approximation to direct scalar volume rendering. *ACM SIGGRAPH Computer Graphics*, 24(5):63–70, 1990.

53 Jos Stam. Multiple scattering as a diffusion process. In Patrick Hanrahan and Werner Purgathofer, editors, *Proc. Eurographics Workshop on Rendering Techniques*, pages 41–50. Springer, 1995.

54 A. J. Stewart. Vicinity shading for enhanced perception of volumetric data. In *Proc. IEEE Visualization*, pages 355–362, 2003.

55 Lifeng Wang, Wenle Wang, Julie Dorsey, Xu Yang, Baining Guo, and Heung-Yeung Shum. Real-time rendering of plant leaves. *ACM Transactions on Graphics (SIGGRAPH '2005)*, 24(3):712–719, 2005.

56 Manfred Weiler, Martin Kraus, Markus Merts, , and Thomas Ertl. Hardware-based ray casting for tetrahedral meshes. In *Proc. IEEE Visualization*, pages 333–340, 2003.

57 David Wiley, Hank Childs, Bernd Hamann, and Kenneth Joy. Ray casting curved-quadratic elements. In *Proc. Eurographics/IEEE TCVG Data Visualization*, pages 201–209, 2003.

58 Peter Williams and Nelson Max. A volume density optical model. In *Proceedings, ACM SIGGRAPH Workshop on Volume Visualization*, pages 61–68, 1992.

59 Peter Williams, Nelson Max, and Clifford Stein. A high accuracy renderer for unstructured data. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):37–54, 1998.

60 Brian Wylie, Kenneth Moreland, Lee Ann Fisk, and Patricia Crossno. Tetrahedral projection using vertex shaders. In *Proceedings, ACM SIGGRAPH Workshop on Volume Visualization*, pages 7–12, 2002.

61 G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, 1982.

62 Ciaxia Zhang, Daqing Xue, and Roger Crawfis. Light propagation for mixed polygonal and volumetric data. In *Proceedings, Computer Graphics International*, pages 249–256. IEEE, 2005.

63 S. W. Zucker and R. A. Hummel. A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):324–331, 1981.