Volume Rendering

Mathematical Models and Algorithmic Aspects

H.C. Hege T. Höllerer D. Stalling

Abstract

In this paper various algorithms for rendering gaseous phenomena are reviewed. In computer graphics such algorithms are used to model natural scenes containing clouds, fog, flames and so on. On the other hand displaying three dimensional scalar datasets as cloudy objects has become an important technique in scientific visualization. Our emphasis is on this latter subject of so-called *direct volume rendering*. All algorithms will be discussed within the framework of linear transport theory. The equation of transfer is derived. This equation is suitable to describe the radiation field in a participating medium where absorption, emission, and scattering of light can occur. Almost all volume rendering algorithms can be shown to solve special cases of the equation of transfer. Related problems like the mapping from data values to model parameters or possible parallelization strategies will be discussed as well.

1 Introduction

One of the most challenging tasks in scientific visualization is the display of volumetric, higher dimensional datasets. Such datasets are being produced in increasing number, size, and complexity in many scientific and engineering disciplines. Examples are simulations in computational fluid dynamics, material sciences, or environmental sciences. In medicine there are also large volumetric datasets arising from various kinds of tomographic devices. In all these fields suitable visualization is very important and necessary in order to understand and analyse the data.

Direct volume rendering or just *volume rendering* is a modern technique for displaying volumetric datasets, especially three dimensional scalar data fields. Traditionally one would visualize such data fields by introducing surface primitives, for example by calculating isovalues or selecting cutting planes. In doing so only a very limited part of the data can be viewed simultaneously. In addition the calculation of isosurfaces is an arbitrarily ill-posed problem in regions where data values do not change very much. This can easily lead to some misinterpretation. Volume rendering in contrast will display the data directly as a transparent, cloudy object. This does not exclude the usage of isosurfaces which play an important role in volume rendering, too. But if the data is not well represented by isosurfaces, there is no need to use them.

Computer graphics algorithms for rendering gaseous phenomena may also be used for realistic image synthesis. Participating media like clouds, fog or flames occur in many natural scenes. In order to get the most realistic results it is important to model these media correctly. However, in this paper we will mainly be concerned with volume rendering as a tool in scientific visualization.

To make the process of image generation well defined, we will restrict ourselves to the camera model of computer graphics, i.e. we will try to reconstruct the image of some scene as it would be recorded by a photographic camera. All physiologic and psychologic effects of image reception, although an important issue of its own, will be ignored in this paper.

The way how an image is formed in a photographic camera can be completely described by the known physical laws of optics. In order to simulate this process, one has to calculate the radiation field as it would be produced by a real scene. Thereby things will be much easier if one ignores the wave character of light and its two possible states of polarization. In fact this approximation is most often used in praxis. The price is that one cannot simulate effects like interference or defraction of light. Neglecting these effects, we are dealing with *geometrical optics*, in contrast to *physical optics* of light.

Considering the approximation of geometrical optics, the interaction of light with surfaces and volume elements can be completely described within the framework of linear transport theory [7, 6]. In the first part of this paper we will discuss the basics of this theory. In the second part we will subsequently discuss various techniques for solving the central equation of transport theory, the equation of transfer. Different solution techniques directly translate into different rendering algorithms. In the third part of this review paper related problems like the mapping from data values to model parameters or possible parallelization strategies will be discussed.

2 Transport Theory of Light

In geometrical optics light can be described by the amount of radiant energy traveling within some frequency interval into a given direction. We will use frequency instead of wavelength because the former remains constant when the index of refraction changes. The interaction of light with surfaces and volume elements is properly described by methods of radiation transport theory. Since in the geometrical optics approximation light does not interact with itself, this theory is a linear one.

In this chapter we will discuss the basics of radiation transport theory. An important point will be the derivation of the equation of transfer. This equation is the basis for all rendering algorithms discussed in the second part of this paper.

2.1 Radiometric Concepts

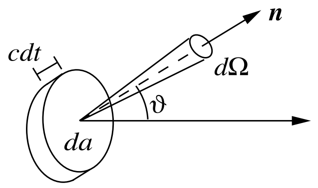
Radiometry deals with the description of light on a transport theoretic level. In this section we will introduce some of the basic terms and notations commonly used in this field.

The central quantity in radiometry is the *specific intensity* $I(\boldsymbol{x}, \boldsymbol{n}, \nu)$. It completely describes the radiation field at any point $\boldsymbol{x}$, giving both its distribution in angle and frequency. Of course the radiation field has a time dependence, too. However, in practice we are mostly interested in steady state solutions, because velocity of light is so large that usually, for example when turning on light in a dark room, the equilibrium state is attained almost instantaneously. Simulating the relaxation process of light would be possible, of course. But most likely such a simulation merely would appear confusing, rather than revealing any useful information. Therefore in this paper we will assume the radiation field to be time-independent.

The amount of radiant energy δE, traveling in time dt within a frequency interval $d\nu$ around ν through a surface element da into an element of solid angle $d\Omega$ in direction $\boldsymbol{n}$, is given by

$$\delta E = I(\boldsymbol{x}, \boldsymbol{n}, \nu)\, \cos\vartheta\, da\, d\Omega\, d\nu\, dt. \tag{1}$$

Here ϑ is the angle between $\boldsymbol{n}$ and the normal on da. More technically the specific intensity I also is called *radiance*. It is measured in units of Watts per meter squared per solid angle per frequency. The factor $\cos\vartheta$ in (1) takes into account area foreshortening.



Alternatively the radiation field can be described by the *photon number density* $\psi(\boldsymbol{x}, \boldsymbol{n}, \nu)$. The number of photons per unit volume at position $\boldsymbol{x}$ in a frequency interval $d\nu$ around ν, traveling with velocity c into an element of solid angle $d\Omega$ in direction $\boldsymbol{n}$, is given by $\psi(\boldsymbol{x}, \boldsymbol{n}, \nu)\, d\Omega\, d\nu$. The number of photons passing a surface da in a time dt therefore is $\psi\,(\cos\vartheta\, da)(c\, dt)(d\Omega\, d\nu)$. Now each photon exactly carries energy $h\nu$, where h is Planck's constant. It follows, that the energy transported through da is given by

$$\delta E = ch\nu\, \psi(\boldsymbol{x}, \boldsymbol{n}, \nu)\, \cos\vartheta\, da\, d\Omega\, d\nu\, dt. \tag{2}$$

Comparing this expression with equation (1), we find that specific intensity is related to photon number density via

$$I(\boldsymbol{x}, \boldsymbol{n}, \nu) = ch\nu\, \psi(\boldsymbol{x}, \boldsymbol{n}, \nu). \tag{3}$$

In this paper we will use specific intensity exclusively.

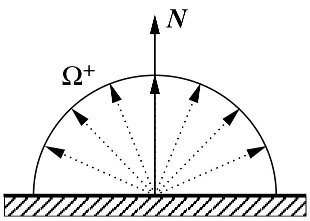
In order to reconstruct the image of a scene as recorded by a camera, we have to compute $I(\boldsymbol{x}, \boldsymbol{n}, \nu)$ for all points being focussed onto the image plane with $\boldsymbol{n}$ pointing towards the camera. For scenes containing participating media, we imagine the scene being fully enclosed by some surface, on which the specific intensity has to be computed. In appendix A we will show that specific intensity is directly related to the brightness of a pixel in the image plane of the camera.

We are now going to discuss some other radiometric quantities. The *radiant flux* or *radiant power* ϕ is defined as the amount of radiant energy transported per second,

$$\phi = dE/dt. \tag{4}$$

Point light sources often are described by their *radiant intensity* J, defined as the radiant flux emitted into an element of solid angle,

$$J = d\phi/d\Omega. \tag{5}$$



A quantity useful for describing extended light sources is the *exitant flux density* or *radiosity* B, defined as the total amount of radiant flux emitted by some surface element da. For opaque surface elements we get this number by integrating specific intensity over the upper hemisphere. A cosine term has to be included to take area foreshortening into account,

$$B = d\phi^{(\text{out})}/da = \int_{\Omega^+} I \cos\vartheta \, d\Omega. \tag{6}$$

In an analogous way the *incident flux density* or *irradiance* D can be defined. We simply have to integrate over the lower hemisphere,

$$D = d\phi^{(\text{in})}/da = \int_{\Omega^-} I \cos\vartheta \, d\Omega. \tag{7}$$

All quantities defined so far were spectral or frequency dependent quantities. We may emphasize this by writing a subscript ν, for example I_ν, instead of just I. Often one is not interested in the exact distribution of radiation in frequency. In this case spectral quantities may be integrated over some relevant frequency range. To emphasize the difference, one also speaks of e.g. *integrated* intensity I or *integrated* radiosity B.

The different radiometric quantities are summarized in Table 1.

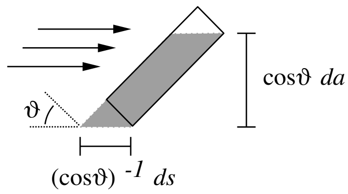
Quantity	Definition	Units
Radiant energy	E	J
Radiant flux, radiant power	$\phi = dE/dt$	W
Radiant intensity	$J = d\phi/d\Omega$	$\mathrm{W\,sr^{-1}}$
Exitant flux density, radiosity	$B = d\phi^{(\mathrm{out})}/da$	$\mathrm{W\,m^{-2}}$
Incident flux density, irradiance	$D = d\phi^{(\mathrm{in})}/da$	$\mathrm{W\,m^{-2}}$
Specific intensity, radiance	$I = dJ/\cos\vartheta\, da$	$\mathrm{W\,m^{-2}\,sr^{-1}}$

Table 1: *Radiometric quantities. The units refer to the integrated form. Spectral quantities receive an additional* Hz^{-1}. *In this case often a subscript* ν *is used to emphasize frequency dependence.*

2.2 Absorption, Emission, and Scattering

When radiation passes through material, energy is generally removed from the beam. We describe this loss in terms of an *extinction coefficient* or *total absorption coefficient* $\chi(\boldsymbol{x}, \boldsymbol{n}, \nu)$. The amount of energy removed from a beam with specific intensity $I(\boldsymbol{x}, \boldsymbol{n}, \nu)$, when passing through a cylindrical volume element of length ds and cross section da, is given by

$$\delta E^{(\mathrm{ab})} = \chi(\boldsymbol{x}, \boldsymbol{n}, \nu) I(\boldsymbol{x}, \boldsymbol{n}, \nu)\, ds\, da\, d\Omega\, d\nu\, dt. \tag{8}$$



Notice that no cosine term appears in this expression. This is because the absorbing volume does not depend on the incident angle, as shown in the figure. The absorption coefficient generally is a function of $\boldsymbol{x}$, $\boldsymbol{n}$, and ν. In practice however, one almost ever deals with the isotropic case, when there is no dependence on $\boldsymbol{n}$. The absorption coefficient is measured in units of m^{-1}. The term $1/\chi$ also is known as the *mean free path* of photons of frequency ν in material. It defines a characteristic length scale for each problem.

The *emission coefficient* $\eta(\boldsymbol{x}, \boldsymbol{n}, \nu)$ is defined in such a way, that the amount of radiant energy within a frequency interval $d\nu$ emitted in time dt by a cylindrical volume element of length ds and cross section da into a solid angle $d\Omega$ in a direction $\boldsymbol{n}$ is

$$\delta E^{(\mathrm{em})} = \eta(\boldsymbol{x}, \boldsymbol{n}, \nu)\, ds\, da\, d\Omega\, d\nu\, dt. \tag{9}$$

It is important to distinguish between true or thermal absorption and emission processes, and the process of scattering. In the former case, energy removed from the beam is converted into material thermal energy, and energy is emitted into the beam at the expense of material energy respectively. In contrast, in a scattering process a photon interacts with a scattering center and emerges from the event moving in a different direction, in general with a different frequency, too. If frequency doesn't change, one speaks of *elastic scattering*, otherwise of *inelastic scattering*.

It is thus convenient to define a *true absorption coefficient* $\kappa(\boldsymbol{x}, \boldsymbol{n}, \nu)$ and a *scattering coefficient* $\sigma(\boldsymbol{x}, \boldsymbol{n}, \nu)$. The total absorption coefficient then is

$$\chi = \kappa + \sigma. \tag{10}$$

The ratio of scattering coefficient and total absorption coefficient, σ/χ, is called *albedo*. An albedo of one means, that there will be no true absorption at all. This is the case of *perfect scattering*.

In an analogous way we break the total emission coefficient into a thermal part or source term $q(\boldsymbol{x}, \boldsymbol{n}, \nu)$ and a scattering part $j(\boldsymbol{x}, \boldsymbol{n}, \nu)$,

$$\eta = q + j. \tag{11}$$

In order to take into account the angle dependence of scattering, we introduce a *phase function* $p(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}', \nu, \nu')$. The amount of radiant energy scattered from frequency ν to frequency ν' and from direction $\boldsymbol{n}$ to direction $\boldsymbol{n}'$, is given by

$$\delta E^{(\mathrm{scat})} = \sigma I \, ds \, da \, d\Omega \, d\nu \, dt \;\times\; \frac{1}{4\pi} p(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}', \nu, \nu') \, d\Omega' \, d\nu'. \tag{12}$$

We assume the phase function to be normalized as follows:

$$\frac{1}{4\pi} \iint p(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}', \nu, \nu') \, d\Omega' \, d\nu' = 1 \tag{13}$$

It should be mentioned that there is no deeper meaning behind the factor $1/4\pi$. It simply cancels the factor 4π resulting from integrating a unity phase function over the sphere.

To get the total amount of radiant energy due to scattering in direction $\boldsymbol{n}'$, we integrate over all possible incident directions $\boldsymbol{n}$ and frequencies ν. By doing so we find that the scattering part of the emission coefficient equals to

$$j(\boldsymbol{x}, \boldsymbol{n}', \nu') = \frac{1}{4\pi} \iint \sigma(\boldsymbol{x}, \boldsymbol{n}, \nu) \, p(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}', \nu, \nu') \, I(\boldsymbol{x}, \boldsymbol{n}, \nu) \, d\Omega \, d\nu. \tag{14}$$

For elastic scattering processes the phase function reduces to

$$p(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}', \nu, \nu') = \delta(\nu - \nu') \, p^{(\mathrm{el})}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}'). \tag{15}$$

Notice that we have assumed the frequency distribution of scattering to be constant for all frequencies, since $p^{(\mathrm{el})}$ does not depend on ν anymore. Most often in practice inelastic scattering is not considered. Then, when talking about the phase function p, what is really meant is $p^{(\mathrm{el})}$.

Many phase functions of interest only depend on $\cos\theta$ instead of $\boldsymbol{n}$ and $\boldsymbol{n'}$, where θ is the angle between these two directions. This restriction means, that scattering takes place in an isotropic medium. The most simple phase function is

$$p = \text{constant} = 1. \tag{16}$$

In this case radiation is scattered equally in all directions.

Another important example of a phase function is the one resulting from *Rayleigh scattering* [7],

$$p = \frac{3}{4}(1 + \cos^2\theta). \tag{17}$$

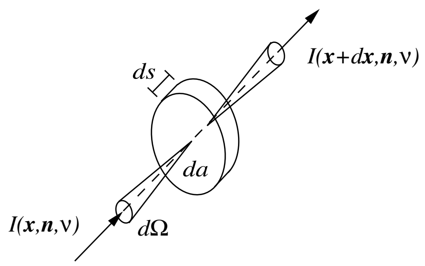
It is easily shown that this function satisfies the normalization condition from equation (13). Rayleigh scattering is a valid approximation for scattering of light at particles much smaller than the wavelength of light. A characteristic feature of Rayleigh scattering is, that there is no preference between forward and backward scattering.

When particles are large compared to the wavelength, then the so-called Mie theory applies. In this case forward scattering strongly dominates. Because Mie theory is quite complicated, often empirical phase functions adapted to experimental results are used [2, 3, 31]. In particular a popular choice is a family of phase functions according to Henvey and Greenstein [13], namely

$$p = \frac{1 - k^2}{(1 + k^2 - 2k\cos\theta)^{3/2}}, \quad k \in (-1, 1). \tag{18}$$

Varying the parameter k provides a continuum between forward scattering ($k > 0$), isotropic scattering ($k = 0$), and backward scattering ($k < 0$).

2.3 Derivation of the Equation of Transfer



The equation of transfer describes the change of specific intensity due to absorption, emission, and scattering. With all material constants given, the radiation field can be calculated from this equation. Consider a cylindrical volume element as shown in the figure. The difference between the amount of energy emerging at position $\boldsymbol{x}+d\boldsymbol{x}$ and the amount of energy incident at $\boldsymbol{x}$ must be equal to the difference between the energy created by emission and the energy removed by absorption. Thus we have

$$\begin{aligned}&\Big\{I(\boldsymbol{x},\boldsymbol{n},\nu) - I(\boldsymbol{x}+d\boldsymbol{x},\boldsymbol{n},\nu)\Big\}\, da\, d\Omega\, d\nu\, dt \\ &\quad = \Big\{-\chi(\boldsymbol{x},\boldsymbol{n},\nu)I(\boldsymbol{x},\boldsymbol{n},\nu) + \eta(\boldsymbol{x},\boldsymbol{n},\nu)\Big\}\, ds\, da\, d\Omega\, d\nu\, dt.\end{aligned} \tag{19}$$

By writing $d\boldsymbol{x} = \boldsymbol{n}\, ds$ we immediately obtain the time independent equation of transfer,

$$\boldsymbol{n}\cdot\boldsymbol{\nabla} I = -\chi\, I + \eta, \tag{20}$$

where we have used the directional derivative

$$\boldsymbol{n}\cdot\boldsymbol{\nabla} I = n_x \frac{\partial I}{\partial x} + n_y \frac{\partial I}{\partial y} + n_z \frac{\partial I}{\partial z} = \lim_{\Delta s \to 0} \frac{I(\boldsymbol{x}) - I(\boldsymbol{x} + \boldsymbol{n}\Delta s)}{\Delta s} \tag{21}$$

Notice that the emission coefficient in general contains a scattering part and thus an integral over I itself. This makes the transport equation be an *integro-differential equation* instead of a simple differential equation. Written out completely it reads

$$\boldsymbol{n}\cdot\boldsymbol{\nabla} I = -(\kappa+\sigma)\, I + q + \frac{1}{4\pi} \iint \sigma(\boldsymbol{x}, \boldsymbol{n}', \nu')\, p(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}, \nu', \nu)\, I(\boldsymbol{x}, \boldsymbol{n}', \nu')\, d\Omega'\, d\nu' \tag{22}$$

or simply

$$\boldsymbol{n}\cdot\boldsymbol{\nabla} I = -(\kappa + \sigma)\, I + q + \frac{1}{4\pi} \int \sigma(\boldsymbol{x}, \boldsymbol{n}')\, p(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n})\, I(\boldsymbol{x}, \boldsymbol{n}')\, d\Omega', \tag{23}$$

if one ignores frequency dependence and thereby inelastic scattering as well.

2.4 Boundary Conditions

The equation of transfer alone does not describe the radiation field completely. Like for other differential equations we have to specify some boundary conditions, too. This is necessary in order to eliminate the constant terms arising from the integration of the gradient operator in (20).

The equation of transfer is only valid away from boundary surfaces. At the surfaces, collectively denoted S, we need to specify what happens. For opaque surfaces boundary conditions are easily to specify. We assume that the surface normal $\boldsymbol{N}$ is always pointing into the volume where the radiation field is present.

In the most simple case we have *explicit boundary conditions*,

$$I(\boldsymbol{x}, \boldsymbol{n}, \nu) = E(\boldsymbol{x}, \boldsymbol{n}, \nu), \quad \boldsymbol{x} \in S \text{ and } \boldsymbol{n} \in \Omega^+ = \{\boldsymbol{n} \,|\, \boldsymbol{n} \cdot \boldsymbol{N} > 0\}. \tag{24}$$

The value of the intensity radiating into the volume is given by a surface emission function E. Like in equation (6), Ω^+ denotes the set of all directions pointing into the volume.

Explicit boundary conditions are independent of I itself. In contrast *implicit* or *reflecting boundary conditions* are defined as

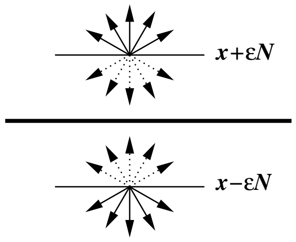
$$I(\boldsymbol{x}, \boldsymbol{n}, \nu) = \iint_{\Omega^-} k(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}, \nu', \nu)\, I(\boldsymbol{x}, \boldsymbol{n}', \nu')\, d\Omega'\, d\nu', \quad \boldsymbol{x} \in S \text{ and } \boldsymbol{n} \in \Omega^+. \tag{25}$$

Here we have introduced the *surface scattering kernel* k. In practice often inelastic scattering is not considered. In this case k can be decomposed into a delta function and an elastic surface scattering kernel, as done with the phase function in equation (15),

$$k(\boldsymbol{x},\boldsymbol{n},\boldsymbol{n}',\nu,\nu') = \delta(\nu-\nu')\, k^{(\mathrm{el})}(\boldsymbol{x},\boldsymbol{n},\boldsymbol{n}'). \tag{26}$$

Of course there may also be a combination of explicit and reflective boundary conditions as well.

If transparent surfaces are included, there is no natural partitioning of directions into Ω^+ and Ω^-. This is, because there is a radiation field present on both sides of transparent surfaces. Light incident on a transparent surface is defracted and changes direction abruptly. Thus specific intensity is not a continuous function for $\boldsymbol{x} \in S$. To define I uniquely we use the following convention,



$$I(\boldsymbol{x},\boldsymbol{n},\nu) = \begin{cases} I(\boldsymbol{x}+\varepsilon\boldsymbol{N},\boldsymbol{n},\nu), & \text{for } \boldsymbol{n}\cdot\boldsymbol{N} \geq 0 \\ I(\boldsymbol{x}-\varepsilon\boldsymbol{N},\boldsymbol{n},\nu), & \text{for } \boldsymbol{n}\cdot\boldsymbol{N} < 0 \end{cases} \quad \boldsymbol{x} \in S \text{ and } \varepsilon \to 0. \tag{27}$$

In this definition we have selected all outgoing radiation. In an analogous way we can also select all radiation incident on a surface element, defining a special quantity $I^{(\mathrm{in})}$ by

$$I^{(\mathrm{in})}(\boldsymbol{x},\boldsymbol{n},\nu) = \begin{cases} I(\boldsymbol{x}+\varepsilon\boldsymbol{N},\boldsymbol{n},\nu), & \text{for } \boldsymbol{n}\cdot\boldsymbol{N} < 0 \\ I(\boldsymbol{x}-\varepsilon\boldsymbol{N},\boldsymbol{n},\nu), & \text{for } \boldsymbol{n}\cdot\boldsymbol{N} \geq 0 \end{cases} \quad \boldsymbol{x} \in S \text{ and } \varepsilon \to 0. \tag{28}$$

We now can express the boundary conditions for transparent surfaces, thereby combining explicit and implicit conditions in one equation:

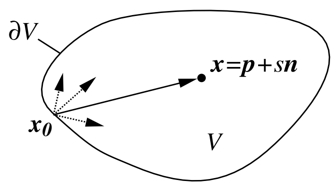
$$I(\boldsymbol{x},\boldsymbol{n},\nu) = E(\boldsymbol{x},\boldsymbol{n},\nu) + \iint k(\boldsymbol{x},\boldsymbol{n}',\boldsymbol{n},\nu',\nu)\, I^{(\mathrm{in})}(\boldsymbol{x},\boldsymbol{n}',\nu')\, d\Omega'\, d\nu', \quad \boldsymbol{x} \in S. \tag{29}$$

If there is no frequency dependence we are left with

$$I(\boldsymbol{x},\boldsymbol{n}) = E(\boldsymbol{x},\boldsymbol{n}) + \int k(\boldsymbol{x},\boldsymbol{n}',\boldsymbol{n})\, I^{(\mathrm{in})}(\boldsymbol{x},\boldsymbol{n}')\, d\Omega', \quad \boldsymbol{x} \in S. \tag{30}$$

In the absence of a participating medium this equation reduces to the well-known *rendering equation*, which is the basis for all surface rendering, as pointed out by Kajiya [21]. Here, in the more general case, it merely serves as a boundary condition for the equation of transfer. The rendering equation will be discussed in more detail in section 2.6.

2.5 Integral Form of the Equation of Transfer



In this section we derive an alternate form of the equation of transfer that is a pure integral equation instead of an integro-differential equation as in (22). We do this by integrating (22) along a ray until a boundary point is reached. First, notice that the operator $\boldsymbol{n}\cdot\boldsymbol{\nabla}$ is the directional derivative along a line $\boldsymbol{x} = \boldsymbol{p} + s\boldsymbol{n}$, with $\boldsymbol{p}$ being some arbitrary reference point. Thus the equation of transfer can be written in a convenient form as

$$\frac{\partial}{\partial s} I(\boldsymbol{x}, \boldsymbol{n}, \nu) = -\chi(\boldsymbol{x}, \boldsymbol{n}, \nu)\, I(\boldsymbol{x}, \boldsymbol{n}, \nu) + \eta(\boldsymbol{x}, \boldsymbol{n}, \nu). \tag{31}$$

The *optical depth* between two points $\boldsymbol{x}_1 = \boldsymbol{p} + s_1\boldsymbol{n}$ and $\boldsymbol{x}_2 = \boldsymbol{p} + s_2\boldsymbol{n}$ is defined as

$$\tau_\nu(\boldsymbol{x}_1, \boldsymbol{x}_2) = \int_{s_1}^{s_2} \chi(\boldsymbol{p} + s'\boldsymbol{n}, \boldsymbol{n}, \nu)\, ds'. \tag{32}$$

We recall that $1/\chi$ is the mean free path of a photon with frequency ν. Therefore we can interpret optical depth as the number of mean free paths between two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Notice that equation (31) has an integrating factor $e^{\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x})}$ and thus can be written as

$$\frac{\partial}{\partial s}\left(I(\boldsymbol{x}, \boldsymbol{n}, \nu)\, e^{\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x})}\right) = \eta(\boldsymbol{x}, \boldsymbol{n}, \nu)\, e^{\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x})}. \tag{33}$$

This equation can be integrated immediately and we obtain

$$I(\boldsymbol{x}, \boldsymbol{n}, \nu)\, e^{\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x})} - I(\boldsymbol{x}_0, \boldsymbol{n}, \nu) = \int_{s_0}^{s} \eta(\boldsymbol{x}', \boldsymbol{n}, \nu)\, e^{\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x}')} ds'. \tag{34}$$

The point $\boldsymbol{x}_0$ is chosen to lie on the boundary surface S. Making use of the fact, that optical depth can be decomposed into

$$\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x}) = \tau_\nu(\boldsymbol{x}_0, \boldsymbol{x}') + \tau_\nu(\boldsymbol{x}', \boldsymbol{x}), \tag{35}$$

we can rewrite equation (34) as follows:

$$I(\boldsymbol{x}, \boldsymbol{n}, \nu) = I(\boldsymbol{x}_0, \boldsymbol{n}, \nu)\, e^{-\tau_\nu(\boldsymbol{x}_0, \boldsymbol{x})} + \int_{s_0}^{s} \eta(\boldsymbol{x}', \boldsymbol{n}, \nu)\, e^{-\tau_\nu(\boldsymbol{x}', \boldsymbol{x})} ds'. \tag{36}$$

This equation can be viewed as the general formal solution of the time-independent equation of transfer. It states that the intensity of radiation traveling along $\boldsymbol{n}$ at point $\boldsymbol{x}$ is the sum of photons emitted from all points along the line segment $\boldsymbol{x}' = \boldsymbol{p} + s'\boldsymbol{n}$, attenuated by the integrated absorptivity of the intervening material, plus an attenuated contribution from photons entering the boundary surface when it is pierced by that line segment. Generally the emission coefficient η will contain I itself, so in fact we have written the transport equation as an integral equation, rather than having obtained a real solution.

2.6 The Rendering Equation

In this section we will shortly discuss an important special case of the transport equation, namely the case of *vacuum conditions*, that is, when there is no absorption, emission, or scattering at all, except on surfaces. Furthermore we will ignore any frequency dependence. The equation of transfer (36) then reduces to

$$I(\boldsymbol{x}, \boldsymbol{n}) = I(\boldsymbol{x}_0, \boldsymbol{n}). \tag{37}$$

This expression states, that specific intensity remains constant along any ray in vacuum. Here $\boldsymbol{x}_0$ is the point where the ray hits some surface element for the first time when traced back. Surface intensity can be determined completely from the boundary conditions. Rays impinging on a surface element at $\boldsymbol{x}$ have to be traced back until some other surface element $\boldsymbol{x}'$ is reached. Thus by substituting

$$I^{(\mathrm{in})}(\boldsymbol{x}, \boldsymbol{n}') = I(\boldsymbol{x}', \boldsymbol{n}'), \tag{38}$$

into equation (30) we obtain the famous *rendering equation* [21]:

$$I(\boldsymbol{x}, \boldsymbol{n}) = E(\boldsymbol{x}, \boldsymbol{n}) + \int k(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n})\, I(\boldsymbol{x}', \boldsymbol{n}')\, d\Omega', \quad \boldsymbol{x} \in S. \tag{39}$$

We may cast the rendering equation into a more familiar form by rewriting the element of solid angle $d\Omega'$ as

$$d\Omega' = \frac{\cos\theta_r'}{|\boldsymbol{x} - \boldsymbol{x}'|}\, da', \tag{40}$$

and decomposing the surface scattering kernel into

$$k(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}) = f_r(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n})\, \cos\theta_i. \tag{41}$$

For naming conventions refer to Figure 1. In the last equation we have introduced the so-called *bidirectional reflectance distribution function* $f_r(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n})$, or BRDF. This function is defined as the fraction of specific intensity dI emerging from a surface into direction $\boldsymbol{n}$ due to some fraction of irradiance dD incident from direction n',

$$f_r(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}) = \frac{dI}{dD} = \frac{dI(\boldsymbol{x}, \boldsymbol{n})}{I^{(\mathrm{in})}(\boldsymbol{x}, \boldsymbol{n}')\, \cos\theta_i\, d\Omega'}. \tag{42}$$

By substituting (40) and (41) into equation (39), thereby transforming the integral over solid angle into an integral over all surface elements visible from $\boldsymbol{x}$, we obtain

$$I(\boldsymbol{x}, \boldsymbol{n}) = E(\boldsymbol{x}, \boldsymbol{n}) + \int f_r(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n})\, \frac{\cos\theta_r'\, \cos\theta_i}{|\boldsymbol{x} - \boldsymbol{x}'|}\, I(\boldsymbol{x}', \boldsymbol{n}')\, da', \quad \boldsymbol{x} \in S. \tag{43}$$

An important special case are diffuse or Lambertian surfaces. Such a surface appears equally bright from all directions, no matter how it is irradiated. Thus

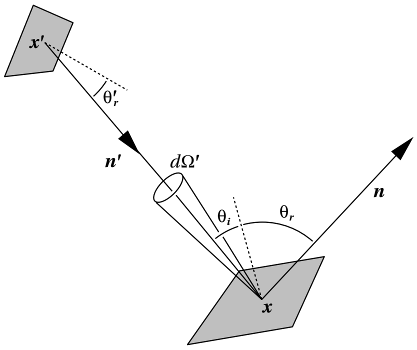


Figure 1: *Vectors and angles appearing in the general surface rendering equation.*

specific intensity $I(\boldsymbol{x})$ is isotropic, i.e. does not depend on direction, and light is equally likely to be scattered in any direction, regardless of the incident direction. In other words, the BRDF itself is isotropic for Lambertian surfaces.

Lambertian surfaces may conveniently be characterized by some reflectance function $\rho(\boldsymbol{x})$, giving the ratio of the overall exitant flux density or radiosity B to the overall incident flux density or irradiance D. The value of the reflectance function is bounded between 0 and 1, where 1 denotes the case of perfect scattering. In particular irradiance is given by

$$D = \int_{\Omega^-} I^{(\mathrm{in})}(\boldsymbol{x}, \boldsymbol{n}') \cos\theta_i \, d\Omega', \tag{44}$$

whereas radiosity is given by

$$B = \int_{\Omega^+} I(\boldsymbol{x}) \cos\theta \, d\Omega = \pi I(\boldsymbol{x}) = \pi \, f_r(\boldsymbol{x}) \int_{\Omega^-} I^{(\mathrm{in})}(\boldsymbol{x}, \boldsymbol{n}') \cos\theta_i \, d\Omega'. \tag{45}$$

Therefore reflectance is related to the BRDF by

$$\rho = B/D = \pi \, f_r. \tag{46}$$

If there are only Lambertian surfaces present, equation (43) reduces to the well known *radiosity equation*,

$$I(\boldsymbol{x}) = E(\boldsymbol{x}) + \frac{1}{\pi}\,\rho(\boldsymbol{x}) \int \frac{\cos\theta_r' \, \cos\theta_i}{|\boldsymbol{x} - \boldsymbol{x}'|} I(\boldsymbol{x}') \, da', \quad \boldsymbol{x} \in S. \tag{47}$$

The standard way to solve this equation is to subdivide surfaces into small patches and to introduce so-called form factors, describing the exchange of radiant energy between every pair of surface patches. One then obtains a huge system of linear equations, describing the radiosity for each patch.

3 Strategies for Solving the Transfer Equation

In the following sections we will discuss various methods for solving the equation of transfer. Most of these methods address special cases of the general transport problem. For example simple ray-integration is only valid if there is no scattering at all. The volume-radiosity method requires isotropic or angle independent absorption, emission and scattering coefficients, and so on. We will discuss all the restrictions and approximations in detail for each method. This section is intended to give an overview over existing solution strategies. It is not our concern to give a complete description of the various algorithms.

3.1 Solving Emission-Absorption Models

In order to make volume-rendering feasible as a generic visualization technique in scientific computing many authors have decided not to model scattering of light at all. The equation of transfer (23) then decouples in ordinate space, and can be solved by simple ray-integration. While this strategy certainly is not suited to model natural phenomena, it nevertheless allows to produce very useful and comprehensive images of 3-dimensional scalar data fields. Because of the enormous size of such datasets ($512 \times 512 \times 512$ samples are not unusual), it is not easy to achieve rendering at nearly interactive rates, even in the case of having no scattering.

Ignoring scattering we are left with the so-called *emission-absorption model*, or *density-emitter model*, a term introduced by Sabella [35]. One can imagine the volume to be filled with small light emitting particles. Rather than being modeled individually, these particles are described by some density function. This is, where the term "density-emitter" comes from. The emission coefficient solely consists of a source term, $\eta = q$. Equivalently the absorption coefficient is $\chi = \kappa$. Because no mixing between different frequencies is possible, we can safely ignore any variable ν from now on.

Let us consider a ray of light traveling along a direction $\boldsymbol{n}$, parametrized by a variable s. The ray enters the volume at position s_0 (compare Figure 2). Suppressing the argument $\boldsymbol{n}$, the equation of transfer in integral form reads

$$I(s) = I(s_0)\, e^{-\tau(s_0,s)} + \int_{s_0}^{s} q(s')\, e^{-\tau(s',s)}\, ds', \tag{48}$$

with optical depth

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds. \tag{49}$$

As usual $I(s_0)$ is given by the boundary conditions.

In order to solve this equation numerically one has to discretize along the ray. If we divide the range of integration into n intervals according to Figure 2, then the specific intensity at position s_k is related to the specific intensity at position s_{k-1} by the identity

$$I(s_k) = I(s_{k-1})\, e^{-\tau(s_{k-1},s_k)} + \int_{s_{k-1}}^{s_k} q(s)\, e^{-\tau(s,s_k)}\, ds. \tag{50}$$

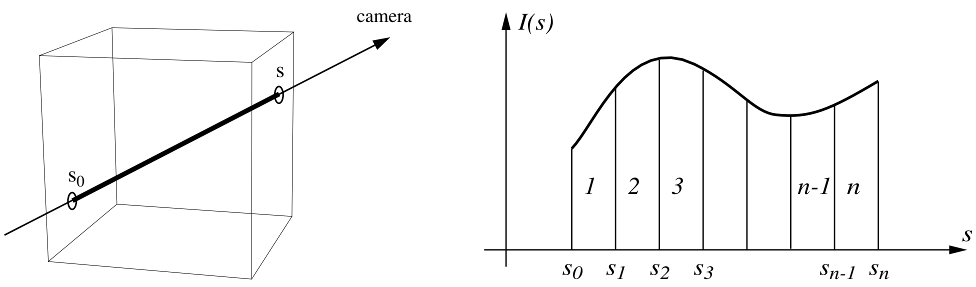


Figure 2: *Naming conventions for ray-integration.*

It is by no means necessary for the s_k to be positioned equidistantly, though this is the most common used procedure. If one knows something about the absorption and emission coefficients, e.g. that they behave like a polynomial of some degree, one can exploit this. This will result in a higher order quadrature rule. However, most often in practice absorption and emission coefficients will be given as an array of data values at discrete points, and fairly little will be known about their functional form.

We introduce the following abbreviations,

$$\theta_k = e^{-\tau(s_{k-1},s_k)} \qquad \text{and} \qquad b_k = \int_{s_{k-1}}^{s_k} q(s)\, e^{-\tau(s,s_k)}\, ds. \tag{51}$$

Specific intensity at s_n now can be written as

$$\begin{aligned} I(s_n) &= I(s_{n-1})\theta_n + b_n = \Big(I(s_{n-2})\theta_{n-1} + b_{n-1}\Big)\theta_n + b_n = \ldots \\ &= \sum_{k=0}^{n} b_k \prod_{j=k+1}^{n} \theta_j, \qquad \text{with} \quad b_0 = I(s_0). \end{aligned} \tag{52}$$

The quantity θ_k is called the *transparency* of the material between s_{k-1} and s_k. Transparency is a number between 0 and 1. An infinitely large optical depth $\tau(s_{k-1}, s_k)$ corresponds to a vanishing transparency $\theta_k = 0$. On the other hand a vanishing optical depth corresponds to full transparency $\theta_k = 1$. Alternatively one often uses *opacity*, defined as $\alpha_k = 1 - \theta_k$.

Equation (52) may be evaluated in two ways. In the following code fragment summation is done from back to front:

$\mathit{intensity} = b_0;$
for $(k = 1;\ k \le n;\ k = k+1)$
$\quad\quad \mathit{intensity} = \theta_k\, \mathit{intensity} + b_k;$

Alternatively summation can be performed from front to back. In this case an additional variable is necessary to hold the accumulated transparency. Summation from front to back has the advantage, that the loop can be terminated, if transparency has become so small, that light from distant volume elements doesn't reach the eye anymore. The corresponding code fragment is:

$intensity = b_n$;
$transparency = \theta_n$;
for $(k = n-1;\ k \geq 0$ **and** $transparency > \epsilon;\ k = k-1)$ {
  $intensity = intensity + b_k\, transparency$;
  $transparency = \theta_k\, transparency$; }

In many cases the quantities θ_k and b_k are approximated by some very simple quadrature rules. A popular choice is to use the trapezoidal rule:

$$\theta_k = \exp\left(-\frac{1}{2}\Big(\kappa(s_{k-1}) + \kappa(s_k)\Big)\,\Delta s_k\right) \tag{53}$$

$$b_k = \frac{1}{2}\Big(q(s_{k-1})\,\theta_k + q(s_k)\Big)\,\Delta s_k. \tag{54}$$

which will give an exact answer for θ_k if the absorption coefficient κ varies linearly between s_{k-1} and s_k. In contrast to get an exact answer for b_k, too, it is not sufficient that the emission coefficient varies linearly, since the integrand for b_k contains an exponential factor. However, there is one important special case in which b_k can easily and exactly be determined from transparency θ_k, namely when both absorption and emission coefficients are related to some density function ρ,

$$\kappa = \kappa_0\,\rho \qquad \text{and} \qquad q = q_0\,\rho, \tag{55}$$

with κ_0 and q_0 being constants. An illumination models of this kind is used in [36] for example. First notice that the following identity holds:

$$\frac{d}{ds}\,e^{-\tau(s,s_k)} = \frac{d}{ds}\,e^{-\kappa_0\int_s^{s_k}\rho(s')\,ds'} = \kappa_0\,\rho(s)\,e^{-\tau(s,s_k)} \tag{56}$$

Substituting this into the definition of b_k we obtain

$$\begin{aligned} b_k = q_0 \int_{s_{k-1}}^{s_k} \rho(s)\,e^{-\tau(s,s_k)}\,ds &= \frac{q_0}{\kappa_0}\int_{s_{k-1}}^{s_k}\frac{d}{ds}\left(e^{-\tau(s,s_k)}\right)\,ds \\ &= \frac{q_0}{\kappa_0}\left(1 - e^{-\tau(s_{k-1},s_k)}\right) = \frac{q_0}{\kappa_0}\,(1-\theta_k). \end{aligned} \tag{57}$$

This is an exact relationship. Inserting this expression into equation (52), all but the first and the last term in the sum cancel each other out. In particular we have

$$I(s_n) = \frac{q_0}{\kappa_0}\,(1 - \theta_0\theta_1\theta_2\cdots\theta_n) = \frac{q_0}{\kappa_0}\,(1 - e^{-\tau(s_0,s_n)}). \tag{58}$$

Notice that now only one exponential per pixel has to be evaluated, instead of one exponential per sampling point. In addition traversing the ray from front to back or vice versa is not necessary anymore when calculating specific intensity.

However, it appears that keeping κ_0 and q_0 constant within the whole volume is too restrictive for most applications. Therefore a common alternative is to allow these constants to take different values in each voxel [30]. In this case expression (57) still remains valid, but when calculating pixel intensity terms do not cancel out anymore like in equation (58). Instead the more general equation (52) has to be used, which requires an ordered traversal along the ray.

3.2 Ray-Casting versus Projection Methods

In principle from the discussion in the previous section it is clear how to solve the equation of transfer for an emission-absorption model without any scattering. However, algorithmic approaches to ray-integration can be divided into two quite different classes, which we will discuss in some detail now.

On the one hand there is the class of so-called *ray-casting algorithms*. Here we run through all pixels of the final image. For every pixel at least one ray is sent into the volume, and intensity is integrated along each ray subsequently. Usually the sampling points s_k are chosen equidistantly. Krüger suggests to take the distance Δs to be half of the lattice spacing [23]. For every sample one has to find the voxel which contains the corresponding point. The data values for each sample usually are obtained by some sort of interpolation from the corner nodes. For tetrahedral volume elements there is a unique linear interpolation formula,

$$f(x, y, z) = a_1 + a_2\,x + a_3\,y + a_4\,z, \tag{59}$$

with the four unkowns a_i being determined from the data values at the four corner nodes of the tetrahedron. For rectilinear volume elements having eight corner nodes one often uses trilinear interpolation,

$$f(x, y, z) = a_1 + a_2\,x + a_3\,y + a_4\,z + a_5\,xy + a_6\,yz + a_7\,zx + a_8\,xyz. \tag{60}$$

However, it should be mentioned that this interpolation scheme is not invariant under rotations. It depends on how a voxel is oriented with respect to the coordinate axes. In some cases also higher order interpolation has been used.

A major problem for ray-casting algorithms is that aliasing can easily occur. It is even not guaranteed that every voxel is hit by some viewing ray. This problem is traditionally solved by *super-sampling*, that is for each pixel more than one ray is casted into the volume. Then an averaged intensity is assigned to the pixel. There has also been work on *adaptive super-sampling* [32] and *pyramidal volume sampling* [36].

When casting rays into a volume, usually a large amount of work has to be repeated. Rays from neighboring pixels often will hit the same volume elements, but it is difficult to make use of that coherence. Data caching mechanisms may be used, so that the interpolation constants in (59) or (60) have to be computed only once. But at least the test for intersection with voxel elements has to be done for each ray separately.

In contrast to ray-casting algorithms data coherence is exploited to a higher degree by *direct projection methods*. Instead of running through all pixels here the volume elements are processed subsequently. Each voxel is projected into the image plane, and intensities are accumulated step by step. This requires the voxels to be sorted from back to front or vice versa, except if emission and absorption coefficients are chosen to be proportional like in equation (55). However, in three dimensional space it may happen that such an ordering is not possible, because three or more polyhedra partially cover each other. It is not known, how often this occurs in typical scientific datasets [46]. Anyway, the problem can be circumvented by dividing the polyhedra into smaller pieces. Beside this it can be shown, that many important lattice types can always be ordered appropriately, among them rectilinear lattices or lattices resulting from a Delaunay tetrahedrization [30].

The projection approach suffers less from aliasing problems, since all data nodes are taken into account. The most interesting feature of projection methods is, that they can be modified to make efficient use of modern graphics hardware. Such hardware is able to display transparent polygons with color C and opacity α very fast. During rendering for each pixel color is updated according to

$$C_{\mathrm{new}} = \alpha\, C + (1-\alpha)\, C_{\mathrm{old}} \tag{61}$$

This may be combined with an interpolation scheme similar to Gouraud shading. If color and opacity are defined for each vertex of a polygon, then the graphics hardware automatically interpolates these values between the vertices. To see how this can be exploited for volume rendering, we recall equation (52), relevant for solving emission-absorption models,

$$I(s_k) = b_k + \theta_k\, I(s_{k-1}), \tag{62}$$

This is exactly the same as equation (61) if we identify $I(s_k) = C_{\mathrm{new}}$, $b_k = \alpha\, C$, $\theta_k = 1-\alpha$, and $I(s_{k-1}) = C_{\mathrm{old}}$. If emission and absorption coefficients are constant within a voxel, then equation (57) is valid. In this case color C is simply proportional to the emission coefficient or source term q_0.

Shirley and Tuchman suggested a volume rendering algorithm which exactly utilizes the interpolation capabilities mentioned above [40]. Their algorithm requires the volume to be made up of tetrahedron elements. The projection of each tetrahedron into the frame buffer is decomposed into a number of triangles. All triangles have one vertex in common, corresponding to the thickest point somewhere in the middle of the projected area. Only for this point ray-integration is performed, yielding values b_k and θ_k or C and α respectively. For the surrounding vertices opacity is set to zero, and a color $C = q(s)$ is used. Then all pixels covered by the projection are rendered using Gouraud alpha shading. This certainly is only a crude approximation to the voxel's exact contribution. Nevertheless, pictures produced in this way look quite appealing.

Recently an even cruder approximation has become popular, known as *splatting* [24, 42]. In this technique even the exact extent of the voxels projection onto the image plane is not computed. Instead every volume element is represented by a user defined transparent surface primitive. What kind of surface primitive is used depends on the computational effort one wants to pay. Relative fast rendering can be accomplished using rectangles with constant color and constant transparency. Polygons with linear or gaussian opacity falloff yield nicer results. However, since also the size of the surface primitives is an adjustable parameter, images produced with the splatting technique have to be interpreted with some caution.

3.3 Ray-Tracing with Single Scattering

It is very much harder to solve the equation of transfer (36) if there is a non-vanishing scattering coefficient. This is because the specific intensity at one point depends on the intensity at all other points. If we have two volume elements a and b then there are infinitely many ways for light traveling from a to b. However, if scattering is not too strong then the amount of light traveling directly from a to b is much larger than the amount of light scattered once, which in turn is much larger then the amount of light scattered twice, and so on. Formally we can see this by writing (36) as a linear operator equation:

$$I = I_0 + \hat{M} I \tag{63}$$

Here the inhomogeneous part I_0 is given by

$$I_0 = I(\boldsymbol{x}_0, \boldsymbol{n}) e^{-\tau(\boldsymbol{x}_0, \boldsymbol{x})} + \int_{s_0}^{s} q(\boldsymbol{x}', \boldsymbol{n}) e^{-\tau(\boldsymbol{x}', \boldsymbol{x})} ds'. \tag{64}$$

The integral operator $\hat{M}$ is defined by

$$\hat{M} I = \iint k(\boldsymbol{x}', \boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n})\, I(\boldsymbol{x}', \boldsymbol{n}') d\Omega'\, ds', \tag{65}$$

where we have introduced the integral kernel

$$k(\boldsymbol{x}', \boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}) = \frac{1}{4\pi} \sigma(\boldsymbol{x}', \boldsymbol{n}')\, p(\boldsymbol{x}', \boldsymbol{n}', \boldsymbol{n})\, e^{-\tau(\boldsymbol{x}', \boldsymbol{x})}. \tag{66}$$

Now if scattering is not too strong we can expand the formal solution of (63) into a Neumann series, getting

$$I = (1 - \hat{M})^{-1} I_0 = \sum_{i=0}^{\infty} \hat{M}^i I_0. \tag{67}$$

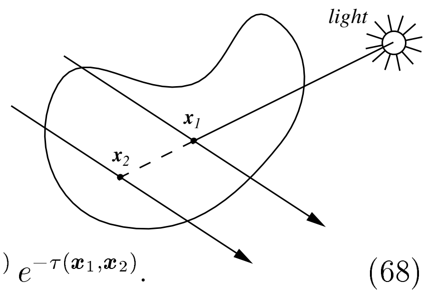
In this expansion different powers of $\hat{M}$ correspond to multiple scattering events, for example $i = 1$ corresponds to single scattering, $i = 2$ correponds to double scattering, and so on.

The formal criterion for expansion (67) to be valid is that the spectral radius of $\hat{M}$ has to be smaller than 1. Since the spectral radius is defined as the maximum eigenvalue of $\hat{M}$ an equivalent statement is that the radiation field should never get amplified by itself. Obviously this is a very sensible criterion, almost ever fulfilled in realistic scenes, with the possible exception of laser-active media. However, if scattering is strong, convergence of (67) may become arbitrarily bad.

A number of ray-tracing algorithms have been suggested for rendering gaseous phenomena which only take into account single scattering events [20, 15]. Terms $i > 1$ in equation (67) are ignored. This approximation is kown as the *low-albedo approximation*. To see how albedo comes in here, let us assume constant emission and absorption coefficients. We can then introduce a dimensionless variable $t = \chi s$. Subsituting this into $\sum \hat{M}^i I_0$ we get a series in σ/χ, which is just the albedo.

Ray-tracing volume densities is especially easy if there are only point light sources present in the scene, but no volume source term q. While sampling along a ray we then have to shoot from each sampling point secondary rays only towards the light sources, instead of doing the integral $\int d\Omega$ appearing in $\hat{M} I_0$ over the whole sphere. If a light source is located at $\boldsymbol{x}_s$, then a secondary ray just computes the quantity $I_0 = I(\boldsymbol{x}_s)\exp(-\tau(\boldsymbol{x}_s, \boldsymbol{x}))$.

In order to speed up computation a so-called shadow table may be used [10]. By looking at the figure we find that the ray traveling from the light source to point $\boldsymbol{x}_2$ computes almost the same quantity as the ray traveling from the light source to point $\boldsymbol{x}_1$. In particular we have



$$I(\boldsymbol{x}_s)e^{-\tau(\boldsymbol{x}_s,\boldsymbol{x}_2)} = I(\boldsymbol{x}_s)e^{-\tau(\boldsymbol{x}_s,\boldsymbol{x}_1)}\, e^{-\tau(\boldsymbol{x}_1,\boldsymbol{x}_2)}. \tag{68}$$

It is therefore worthwhile not to throw away information previously collected, but to efficiently compute I_0 at a number of selected locations in three dimensional space in a preprocessing step. Instead of shooting secondary rays one just has to look up values in the shadow table now. Lookup generally requires some sort of interpolation, but quality seems to be acceptable even for sparse tables.

3.4 The High-Albedo Approximation

A high-albedo approximation has been discussed by Kajiya and von Herzen [20]. This treatment is rather formal, and as far as we know their formulae never have been used in any implementation. Nevertheless we shortly summarize this approximation for sake of completeness.

A high-albedo approximation is obtained by expanding the specific intensity into a power series in $\beta = (1 - \sigma/\chi)$, where σ/χ is the albedo. We are assuming the emission and absorption coefficients to be proportional to some mass density ρ. That means that we have a constant albedo.

Ignoring frequency dependence, the transport equation (22) can be written as

$$\chi^{-1}\,\boldsymbol{n}\cdot\boldsymbol{\nabla} I = -I + q/\chi + (1-\beta)\,\frac{1}{4\pi}\int p(\boldsymbol{x},\boldsymbol{n}',\boldsymbol{n})\,I(\boldsymbol{x},\boldsymbol{n}')\,d\Omega'. \tag{69}$$

In operator notation we have

$$\hat{L}I + (1-\beta)\hat{M}I = 0, \tag{70}$$

with

$$\hat{L}I = \chi^{-1}\,\boldsymbol{n}\cdot\boldsymbol{\nabla} I + I - q/\chi \tag{71}$$

$$\hat{M}I = \frac{1}{4\pi}\int p(\boldsymbol{x},\boldsymbol{n}',\boldsymbol{n})\,I(\boldsymbol{x},\boldsymbol{n}')\,d\Omega'. \tag{72}$$

Using the high-albedo series expansion for I, given by

$$I = \sum_{k=0}^{\infty} \beta^k I_k, \tag{73}$$

and equating equal powers of β, we find a system of coupled equations, namely

$$\begin{aligned} \hat{L}I_0 + \hat{M}I_0 &= 0 \\ \hat{L}I_k + \hat{M}I_k &= \hat{M}I_{k-1}, \quad k \geq 1. \end{aligned} \tag{74}$$

I_0 is the solution for the conservative $\beta = 0$ case when there is perfect scattering. The I_k are corrections relevant for $\beta > 0$. These corrections may be obtained from equation (74), if I_0 is already known.

In the case of perfect scattering the radiation field is likely to be very diffuse and almost equally distributed in all directions. Therefore an expansion of I_0 in terms of spherical harmonics is plausible,

$$I_0(\boldsymbol{x},\boldsymbol{n}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} I^{lm}(\boldsymbol{x})\,Y_{lm}(\vartheta,\phi). \tag{75}$$

Kajiya points out that for purposes of computer graphics the series may be truncated after the *p-wave* or $l = 1$ term.

By substituting equation (75) into equation (69), multipying by $Y^*_{l'm'}$, and integrating over all directions, one obtains a system of coupled first order differential equations for I^{lm}:

$$\begin{aligned} \sum_{l,m} \chi^{-1}\,\boldsymbol{\nabla} I^{lm}(\boldsymbol{x}) \cdot \langle Y_{l',m'}|\boldsymbol{n}|Y_{lm}\rangle \; - \\ I^{lm}(\boldsymbol{x})\,\delta_{ll'}\delta_{mm'} + \chi^{-1}q + I^{lm}(\boldsymbol{x})\,\langle Y_{l',m'}|p|Y_{lm}\rangle = 0 \end{aligned} \tag{76}$$

Here we have written $\langle X|O|Y\rangle$ to denote the inner product integral

$$\int X^* OY \, d\Omega. \tag{77}$$

It is possible to analytically find expressions for the coefficients $\langle Y_{l',m'}|\boldsymbol{n}|Y_{lm}\rangle$ and $\langle Y_{l',m'}|p|Y_{lm}\rangle$. In this way we may solve equation (76) for the functions $I^{lm}(\boldsymbol{x})$, thus obtaining the specific intensity I_0 for the conservative case via equation (75). By calculating the corrections I_k from equation (74) we finally find specific intensity for the case $\beta < 1$.

3.5 The Volume-Radiosity Method

In surface rendering the general rendering equation (43) takes a much simpler form if we assume diffuse reflection for all surface elements. In this case the specific intensity does not depend on direction. We only have to find a single scalar quantity $I(\boldsymbol{x})$ instead of a distribution $I(\boldsymbol{x}, \boldsymbol{n})$ for each point of interest. The rendering equation (43) reduces to the radiosity equation (44).

We may generalize this concept to the case of participating media. Of course we can't assume specific intensity to be isotropic for each point in three dimensional space. But if the coefficient $\eta = q + j$ does not depend on direction, then for each volume element at least the part of intensity due to emission and scattering is isotropically distributed. In this case the equation of transfer in integral form without frequency dependence reads

$$I(\boldsymbol{x}, \boldsymbol{n}) = I(\boldsymbol{x}_0, \boldsymbol{n}) \, e^{-\tau(\boldsymbol{x}_0, \boldsymbol{x})} + \int_{s_0}^{s} \eta(\boldsymbol{x}') \, e^{-\tau(\boldsymbol{x}', \boldsymbol{x})} ds', \tag{78}$$

with the total emission coefficient given by

$$\eta(\boldsymbol{x}) = q(\boldsymbol{x}) + \frac{1}{4\pi} \sigma(\boldsymbol{x}) \int I(\boldsymbol{x}, \boldsymbol{n}) \, d\Omega. \tag{79}$$

Substituting equation (78) into (79) and rewriting the integrals over s' and Ω as a single integral over volume, using

$$ds' \, d\Omega = \frac{1}{|\boldsymbol{x} - \boldsymbol{x}'|^2} \, dV', \tag{80}$$

we obtain an integral equation for $\eta(\boldsymbol{x})$,

$$\begin{aligned} \eta(\boldsymbol{x}) \quad &= \quad q(\boldsymbol{x}) + \frac{1}{4\pi} \sigma(\boldsymbol{x}) \int I(\boldsymbol{x}_0, \boldsymbol{n}) \, e^{-\tau(\boldsymbol{x}_0, \boldsymbol{x})} \, d\Omega \\ &\quad + \frac{1}{4\pi} \sigma(\boldsymbol{x}) \int \frac{\eta(\boldsymbol{x}')}{|\boldsymbol{x} - \boldsymbol{x}'|^2} \, e^{-\tau(\boldsymbol{x}', \boldsymbol{x})} dV'. \end{aligned} \tag{81}$$

The location $\boldsymbol{x}_0$ where the boundary surface is hit depends on $\boldsymbol{x}$ and $\boldsymbol{n}$. To emphasize this, we may explicitly write $\boldsymbol{x}_0 = \boldsymbol{x} + s_0 \boldsymbol{n}$. The intensity $I(\boldsymbol{x}_0, \boldsymbol{n})$ has to be determined from the boundary conditions as usual.

Equation (81) is called the *volume radiosity equation* for the continuous case. It resembles very much the ordinary radiosity equation, and thus can be solved with the same methods. Usually the volume will be divided into small elements, and some functional form of $\eta(\boldsymbol{x})$ over these elements is assumed. In the most simple case a constant approximation $\eta(\boldsymbol{x}) = \eta_{V_i}$ for a volume element V_i is taken. Then equation (81) can be rewritten as a system of linear equations, which is easily solved by standard methods.

In contrast to surface radiosity methods we cannot immediately render images from arbitrary directions after having solved the integral equation for $\eta(\boldsymbol{x})$. Instead we have to perform a ray integration in order to get $I(\boldsymbol{x}, \boldsymbol{n})$ from (78). This can be done with the methods discussed in section 3.1 and 3.2.

We will now shortly discuss how to discretize the volume radiosity equation, thereby following loosely Rushmeier and Torrance [34]. Their treatment is completely analog to the so-called *classical radiosity method* for surfaces, which is build on a piecewise constant approximation of specific intensity. All surfaces are assumed to be diffuse reflecting ones characterized by a directional independent reflectivity function $\rho(\boldsymbol{x})$, compare section 2.6. In this case boundary conditions can be obtained from equation (30) by substituting $k = \frac{1}{\pi}\,\rho\cos\theta_i$. In order to express the intensity $I^{(\mathrm{in})}$ incident on a surface patch the original equation of transfer (78) has to be inserted:

$$\begin{aligned} I(\boldsymbol{x}) &= E(\boldsymbol{x}) + \frac{1}{\pi}\,\rho(\boldsymbol{x}) \int \cos\theta_i\, I^{(\mathrm{in})}(\boldsymbol{x}, \boldsymbol{n})\, d\Omega \\ &= E(\boldsymbol{x}) + \frac{1}{\pi}\,\rho(\boldsymbol{x}) \int \cos\theta_i\, e^{-\tau(\boldsymbol{x}', \boldsymbol{x})}\, I(\boldsymbol{x}')\, d\Omega \\ &\qquad + \frac{1}{\pi}\,\rho(\boldsymbol{x}) \int \frac{\cos\theta_i}{|x - x'|^2}\, e^{-\tau(\boldsymbol{x}', \boldsymbol{x})}\, \eta(\boldsymbol{x}')\, dV', \quad \boldsymbol{x} \in S. \end{aligned} \tag{82}$$

If we assume constant intensity I_{A_i} for sufficiently small surface patches A_i and constant emission η_{V_i} for sufficiently small volume elements V_i, then equations (82) and (81) are readily expressed as a set of coupled linear equations,

$$I_{A_i} = E_{A_i} + \rho_{A_i}\Big(\sum_j F_{A_i A_j}\, I_{A_j} + \sum_k F_{A_i V_k}\, \eta_{V_k}\Big) \tag{83a}$$

$$\eta_{V_i} = q_{V_i} + \sigma_{V_i}\Big(\sum_j F_{V_i A_j}\, I_{A_j} + \sum_k F_{V_i V_k}\, \eta_{V_k}\Big) \tag{83b}$$

In these equations we have introduced various *form factors* F describing the interaction between different surface and volume elements. In particular the *surface-to-surface* form factor $F_{A_i A_j}$ gives the contribution of specific intensity radiated from A_j to A_i. It is defined as

$$F_{A_i A_j} = \frac{1}{\pi}\,\frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_r' \cos\theta_i}{|\boldsymbol{x} - \boldsymbol{x}'|^2}\, e^{-\tau_\nu(\boldsymbol{x}', \boldsymbol{x})}\, da'\, da. \tag{84}$$

This factor differs from the conventional one used in surface radiosity only by the inclusion of the transparency term $\exp(-\tau)$. Notice that the outer integral in this definition results from averaging the inner area-to-point contribution over the receiving patch area A_i. In order to make the form factor symmetric under exchange of A_i and A_j the normalization factor A_i^{-1} is often taken out of the above definition. It then appears in the linear system (83a) instead.

The *surface-to-volume* form factor is defined as

$$F_{A_i V_j} = \frac{1}{4\pi} \frac{1}{A_i} \int_{A_i} \int_{V_j} \frac{\cos\theta_i}{|\boldsymbol{x} - \boldsymbol{x}'|^2} e^{-\tau_\nu(\boldsymbol{x}', \boldsymbol{x})} \, dV' \, da, \tag{85}$$

whereas the *volume-to-surface* factor is given by

$$F_{V_i A_j} = \frac{1}{4\pi} \frac{1}{V_i} \int_{V_i} \int_{A_j} \frac{\cos\theta_r'}{|\boldsymbol{x} - \boldsymbol{x}'|^2} e^{-\tau_\nu(\boldsymbol{x}', \boldsymbol{x})} \, da' \, dV. \tag{86}$$

Again, when the normalization factors A_i^{-1} and V_i^{-1} are taken out of these definitions, there is a symmetry relationship between both factors and we don't have not to distinguish between surface-to-volume and volume-to-surface terms anymore. Note that in this case θ_r' becomes θ_i and vice versa.

Finally the *volume-to-volume* form factor is defined as

$$F_{V_i V_j} = \frac{1}{4\pi} \frac{1}{V_i} \int_{V_i} \int_{V_j} \frac{1}{|\boldsymbol{x} - \boldsymbol{x}'|^2} e^{-\tau_\nu(\boldsymbol{x}', \boldsymbol{x})} dV' \, dV. \tag{87}$$

In the special case of optically thin media, volume-to-volume interactions are quite small and may be ignored at all [33].

The above method has been extended by Bathe and Tokuta [1] to the case of more general anisotropic scatter, using spherical harmonics to approximate the directionally dependent phase function.

4 Related Problems

4.1 Mapping from Data Values to Model Parameters

In this section we will discuss how the data values of some scalar data field $f(\boldsymbol{x})$ are appropriately mapped to model parameters for volume rendering. This is a highly non-trivial problem. Many suggestions have been made, but there is no commonly accepted solution strategy yet. The large number of possiblities is both, a major problem and a big opportunity to find a proper mapping.

Emission-absorption models

We will begin with simple emission-absorption models, as discussed in section 3.1. In these models no scattering is taken into account. Only the emission and absorption coefficients $q(\boldsymbol{x}, \boldsymbol{n}, \nu)$ and $\kappa(\boldsymbol{x}, \boldsymbol{n}, \nu)$ have to be specified. There cannot occur any scattering, in particular no inelastic scattering, so intermixing of light with different frequencies is impossible. Usually the whole spectrum of visible light is approximated by finitely many frequencies for which specific intensity is calculated independently. After a detailed study of experimental data, Meyer [29] concludes that for most applications a set of four carefully chosen frequencies provides a good balance of cost and accuracy. Nevertheless, the most popular color model is still the RGB-model, which relies on only three frequencies, ν_i with $i =$ red, green, blue. The advantage of this model is, that a triple of RGB-intensities can directly be used to display that particluar color on a color CRT.

In the most simple case both emission and absorption coefficient do not depend on direction $\boldsymbol{n}$. Instead of dealing with q and κ one may specify the quantities b_k and θ_k from equation (51) directly. Alternatively color C and opacity α are often used instead. These quantities are proportional to q and κ in some approximation. A high color value corresponds to a high emission coefficient, and a high opacity value corresponds to a high absorption coefficient.

In many existing systems the user has to edit color and opacity maps to provide the mapping from data values to model parameters. Usually the same opacity is used for the red, green, and blue frequencies:

$$\begin{aligned} \text{OpacityMap: } & f(\boldsymbol{x}) \mapsto \alpha \\ \text{ColorMap: } & f(\boldsymbol{x}) \mapsto C_i, \quad i = \text{r, g, b} \end{aligned} \tag{88}$$

A common task is to extract a distinct isovalue surface from a volume dataset. Of course this can be achieved simply by choosing non-zero opacities only in a small window within the opacity map. However, especially for noisy and inhomogeneous datasets more pleasing images are obtained if the thickness of the transition region containing non-zero opacities stays constant throughout the whole volume. Levoy [25] suggests the following formula to obtain an isovalue surface around a data value f_0 with thickness r:

$$\alpha = \alpha_0 \begin{cases} 1 & \text{if } |\boldsymbol{\nabla} f(\boldsymbol{x})| = 0 \text{ and } f(\boldsymbol{x}) = f_0 \\ 1 - \dfrac{1}{r}\dfrac{|f_0 - f(\boldsymbol{x})|}{|\boldsymbol{\nabla} f(\boldsymbol{x})|} & \text{if } |\boldsymbol{\nabla} f(\boldsymbol{x})| > 0 \text{ and } \dfrac{|f_0 - f(\boldsymbol{x})|}{|\boldsymbol{\nabla} f(\boldsymbol{x})|} < r \\ 0 & \text{otherwise.} \end{cases} \tag{89}$$

Notice that the term containing the local gradient vector is just an estimate for $|\boldsymbol{x}_0 - \boldsymbol{x}|$, which is the distance from the current position to some point $\boldsymbol{x}_0$ with $f(\boldsymbol{x}_0) = f_0$. To see this, expand f around $\boldsymbol{x}$ up to first order.

Many datasets consist of regions or compartments characterized by a nearly constant data value and discontinuities at their boundaries. In order to visualize such datasets, different opacities can be assigned to different regions. Often it might be useful to emphasize the region boundary. In this case the original chosen opacities can be scaled by a discrete approximation of the local gradient vector [25].

The local gradient vector also is very useful for determining the color value C. When a simple color map (88) is used, images tend to look quite flat and diffuse. Upson and Keeler [41] incorporate a term simulating a diffuse reflection:

$$C = k_a I_a + k_d \sum_{\text{lights}} \boldsymbol{N} \cdot \boldsymbol{L}_j \, I_j \tag{90}$$

Here k_a and k_d are the ambient and diffuse light coefficients, which can be obtained from the data values using a colormap. I_a is a global ambient light intensity, I_j is the intensity of the j-th point light source. The vector $\boldsymbol{N}$ can be viewed as a local surface normal. It is given by the normalized local gradient vector. Finally $\boldsymbol{L}_j$ gives the direction from the current position towards the j-th light source.

Notice that equation (90) simulates a diffuse reflection by choosing the emission coefficient q or color C to be of a special form. More realistically any reflection should be modeled with an appropriate non-zero scattering coefficient σ. In equation (90) the light source intensity I_j is the same for all locations. It becomes not attenuated by material lying between the light and the current location. One also says, that the effect of *self-shadowing* is disregarded in this case.

Of course it is possible to generalize equation (90) to include other terms from traditional computer graphics, like specular reflection or depth cueing. Levoy [25] for example uses the Phong illumination model described in [4].

Some authors use color more symbolically, thereby deviating more and more from the physical description of light interacting with some medium. Sabella [35] employs the HSV-color model instead of working with the RGB-model. Only the V-component of pixel color, representing *value* or brightness, is obtained from a ray-integration as defined in equation (52). The H-component or *hue* is determined from the peak data value encountered along a ray. Finally the S-component or *saturation* of color is used for depth cueing. It's value is obtained either from the distance at which the peak value was encountered or alternatively by the center of gravity along a ray.

Models with non-zero scattering

In most cases when more sophisticated algorithms have been implemented, taking into account a possible scattering of light, these methods have been used to model natural phenomena and not as a tool in scientific visualization. There is not much experience on how to map scientific data sets to the parameters of a scattering process in a reasonable way. Notice that not just a scattering coefficient σ has to be specified, but also a phase function $p(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{n}')$. Some authors argue that scattering merely introduces confusing artefacts rather than giving any visual cues, and therefore isn't useful in scientific visualization at all.

However, non-zero scattering at least is useful to implement illumination models containing diffuse or specular reflection based on the local gradient vector in a more realistic way than discussed above. This is the only way to correctly take into account self-shadowing of volumes. Especially in scenes containing surface-like structures, self-shadowing might be useful to enhance spatial reception.

Krüger [22] suggests to visualize local fluctuations, deviations, or noise in scientific data sets with the help of a non-zero scattering kernel. When applied correctly, a characteristic local texture can be produced in this way. Regions with small fluctuations should appear rather smooth, whereas noisy parts of the data-set should exhibit some typical structure.

4.2 Simultaneous Display of Non-Volumetric Objects

For many applications in scientific visualization it is important that both volumetric and surface defined objects can be rendered at the same time. This is essential in order to illustrate an image with text and other objects, e.g. coordinate axes. From the mathematical point of view surfaces inside a volumetric object are described through the boundary conditions (compare section 2.4). However, not all algorithms can handle arbitrary surfaces inside a volume, and quite different solutions have been suggested to cope with this problem.

In this section we will again concentrate on emission-absorption models, because these are the most interesting ones for interactive application and have been studied more extensively. An early approach for integrating the display of polygonal defined objects into volume-rendering was to voxelize these objects [16, 17, 18]. The renderer then only has to cope with a single type of objects, namely with volumetric ones. However, the approach strongly suffers from aliasing problems. Smooth surfaces usually exhibit strange artefacts after voxelization. In order to obtain results of reasonable quality, one has to deal with volumes of relatively high resolution, even if the original volume dataset is defined on a much coarser grid.

Another simple technique to simultaneously display volumetric and non-volumetric objects is the so-called Z-buffer merging technique [19]. Here two images are computed independently, one using volume rendering and the other using traditional surfaces rendering. For each image also a Z-buffer is computed. In the case of volume rendering the Z-buffer may either contain the depth of the first non-zero voxel, or the depth at which some user-defined opacity is exceeded. Then both images are combined in a post-processing step, simply taking each pixel from that image having the lowest Z-value. Of course this is a kind of brute-force strategy. Nevertheless, the method might be useful when a polygonal defined object has to be moved interactively in a volume rendered scene. Only surface rendering has to be repeated, whereas the volume rendered image remains the same.

An improvement of the above method was suggested by Levoy [26]. Again a surface rendered image and a corresponding Z-buffer are computed in a preprocessing step. For volume rendering a ray-casting algorithm is employed. Ray-integration is done from front to back, but instead of traversing the whole volume, integration stops, when the depth specified in the Z-buffer is reached. Accumulated opacity from volume rendering determines how much from the color of the underlying polygon is visible in the final image.

The display of surface primitives may more easily be combined with volume rendering, when a projection method is used, which approximates transparent volume elements by transparent surface elements. Such algorithms are described in [30, 40]. Since the renderer has to deal with (transparent) polygons exclusively, the only difficulty is to ensure for the right depth ordering, compare section 3.2. Usually a small error is introduced at locations, where an external polygon intersects a transparent volume element. The only way to circumvent these errors is to subdivide the voxel, so that correct depth ordering becomes possible.

4.3 Parallel Algorithms

Volume rendering is a promising visualization technique, but still suffers from its relatively high computational costs. In order to make the method feasible for interactive applications, several attempts have been made to implement it on parallel computers [28, 38, 37]. Up to now parallel algorithms have been developed for emission-absorption models only. In this case the equation of transfer can be solved by a simple ray-integration. Integrations can be performed independently for each ray, thus the problem should be well suited for a parallel implementation. Massively parallel computers are restricted to have physical distributed memory today. It is not possible to hold the whole volume dataset consisting of 256^3 or even more nodes in the local memory of each processor. Instead the dataset must be splitted, which makes the algorithm more complicated. Up to now most parallel implementations are based upon the ray-casting approach, although Upson and Keeler [41] pointed out, that projection methods should be more amenable for parallelization.

Schröder and Salem [38] discuss a simple parallel implementation on a SIMD-architecture. In a pre-processing step data values are transformed, so that they are aligned in view direction. To every processor a single voxel is assigned. The rotation can be decomposed in such a way, that communication between neighboring processors is only necessary along the main axes. After this, ray-integration can easily be performed in parallel. However, it is not possible to simultaneously scale the dataset with this method.

Schröder and Krüger follow a different approach. Instead of a voxel they assign a viewing ray to each processor of a massively parallel SIMD-computer. Then integration for all pixels is performed in parallel. Depending on the precision of ray-integration and the complexity of the illumination model, they reported computing times varying from 5 to 30 seconds for a medium size dataset of 128^3 nodes.

Corie and Mackerras [8] reported on an implementation of a ray-casting method on a more flexible MIMD-computer. A MIMD-architecture allows to use a single processor much more efficiently. At the same time it is much more difficult to share data and work among the processors in such a way, that on the one hand data coherence can be exploited in order to minimize communication, and on the other hand good load balacing is ensured. Exploiting data coherence means, that a single processor should work on neighboring rays whenever possible. However, computational costs for different rays can vary about several orders of magnitude. The compromise between data coherence and load balancing depends on both, the hardware architecture and the data to visualize.

Due to the high computational costs, the need of interactivity and the inherent parallelism, volume rendering is an ideal task for massively parallel computers. For the near future one can expect a rapid development of parallel volume rendering algorithms, special purpose hardware and powerful implementations that make it possible to visualize large volumetric data sets at interactive rates.

Appendix

A Specific intensity and pixel brightness

In this appendix we will demonstrate that specific intensity is directly related to pixel brightness inside a camera. The following discussion closely follows the one in Horn and Sjoberg [14]. For sake of simplicity we assume a properly focused imaging system. All rays emerging from a certain point in the scene should meet at a single point in the camera. Likewise, all rays emerging from a small surface area da_0 in the scene should be projected into some area da_p in the image plane. No light from other parts of the scene should reach da_p. A simple model of a camera is depicted in Figure 3.

In order to determine the brightness of a pixel one has to consider the exposure of film in the camera, that is, the amount of energy being focused into the area da_p occupied by the pixel. This amount of energy is proportional to irradiance or incident flux density $D = d\phi/da_p$, compare Table 1. On the other hand, the flux

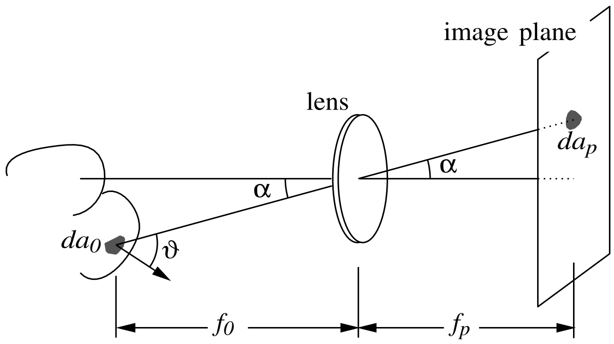


Figure 3: *A simple imaging device. Light collected by the lens from surface area da_0 is projected into some area da_p in the image plane.*

density emitted from da_0, being projected into da_p, is given by

$$d\phi = da_0 \int_{\Omega_a} I \, \cos\theta \, d\Omega. \tag{91}$$

Here the integral is over the solid angle Ω_a subtended by the camera's entrance aperture. Making use of this expression, we obtain

$$D = d\phi/da_p = \frac{da_0}{da_p} \int_{\Omega_a} I \, \cos\theta \, d\Omega. \tag{92}$$

If θ_0 is the angle between the normal on da_0 and the line to the entrance aperture nodal point, while α is the angle between this line and the optical axis, then, by equating solid angles we find

$$\frac{da_0 \, \cos\theta_0}{f_0^2} = \frac{da_p \, \cos\alpha}{f_p^2}. \tag{93}$$

From this relationship we obtain the ratio da_0/da_p. Inserting this into equation (92) yields

$$D = (f_0/f_p)^2 \, \cos\alpha \int_{\Omega_a} I \, \frac{\cos\theta \, d\Omega}{\cos\theta_0}. \tag{94}$$

We assume, that the lens is small compared to the distance f_0. In this case θ and θ_0 are approximately the same, and the cosines in equation (94) cancel. In addition we assume, that specific intensity I is approximately constant within Ω_a, and therefore can be taken out of the integral. If the diameter of the camera's lens is d, then the solid angle Ω_a as seen from da_0 is given by the forshortened area $(\pi/4)\, d^2 \, \cos\alpha$, divided by the distance $f_0/\cos\alpha$ squared. Finally one obtains

$$D = (\pi/4)\,(d/f_p)^2 \, \cos^4\!\alpha \, I. \tag{95}$$

Pixel brightness therefore is proportional to specific intensity. The factor of proportionality depends on α and thus is not constant within the image plane. In the case of vignetting, when the entrance aperture becomes partially occluded in some directions, additional corrections have to be taken into account, also depending on pixel position. Ideally, an imaging device should be calibrated so that sensitivity is the same for the whole picture.

Other kinds of imaging systems, such as microscopes or mechanical scanners, lead to expressions somewhat different from equation (95). Generally, however, pixel brightness or image irradiance is proportional to specific intensity in such systems, too.

B Color Plates

In this section we want to present some results obtained from different volume rendering algorithms.[1] The few examples cannot provide an overview, they merely illustrate some of the capabilities of the volume rendering approach.

In Figure 4 a three dimensional temperature distribution inside the human pelvic region is visualized using the hierarchical splatting algorithm described by Laur and Hanrahan [24]. The splatting technique is based on approximating participating volume elements by simple transparent polygonal shapes, so-called *splats*, compare section 3.2. Because it is impossible to model any kind of scattering, density clouds always appear quite diffuse and fuzzy. An advantage of the splatting approach is that non-volumetric objects like the hipbones in the figure can easily be included.

Figure 5 shows a medical CT-data set visualized using a ray casting algorithm. No surface patches are present in the scenes. Again no scattering is taken into account, but for the bones on the left the emission coefficient was chosen similar to equation (90). In this expression the local gradient vector appears, allowing to simulate the reflection of light on a surface, thereby providing important visual cues. This method is quite simple, but it does not account for self-shadowing of light as discussed in section 4.1.

Real scattering was taken into account in Figure 6. Some lightbeams from the ceiling spots are clearly visible in a smoky room. The image was produced by solving the equation of transfer using a Monte-Carlo based technique. The scene is made up of surface patches and volume elements. Radiant energy is exchanged between these by shooting rays of light. In this way specific intensity can be found for each patch or volume element. For details refer to Shirley [39].

[1] For technical reasons the color plates are placed on the very last page of this report.

C Important Formulae

Absorption coefficient:

$$\chi = \kappa + \sigma$$

χ = total absorption coefficient
κ = true absorption coefficient
σ = scattering coefficient

Emission coefficient:

$$\eta = q + j$$

η = total emission coefficient
q = true emission coefficient
j = scattering part of emission

Scattering part of emission:

$$j(\boldsymbol{x}, \boldsymbol{n}, \nu) = \frac{1}{4\pi} \iint \sigma(\boldsymbol{x}, \boldsymbol{n}', \nu')\, p(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}, \nu', \nu)\, I(\boldsymbol{x}, \boldsymbol{n}', \nu)\, d\Omega'\, d\nu'$$

p = phase function
I = specific intensity

Normalization of phase function:

$$\frac{1}{4\pi} \iint p(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}, \nu', \nu)\, d\Omega\, d\nu = 1$$

Equation of transfer (differential form):

$$\begin{aligned} \boldsymbol{n} \cdot \boldsymbol{\nabla} I &= -\chi I + \eta \\ &= -(\kappa + \sigma) I + q + \frac{1}{4\pi} \iint \sigma(\boldsymbol{x}, \boldsymbol{n}', \nu')\, p(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}, \nu', \nu)\, I(\boldsymbol{x}, \boldsymbol{n}', \nu)\, d\Omega'\, d\nu' \end{aligned}$$

Equation of transfer (integral form):

$$I = I_0\, e^{-\tau(x_0, x)} + \int_{s_0}^{s} (q + j)\, e^{-\tau(x', x)}\, ds'$$

I_0 = specific intensity at boundary surface at $\boldsymbol{x}_0$
$\tau(\boldsymbol{x}_1, \boldsymbol{x}_2)$ = optical depth $= \int_{s_1}^{s_2} \chi(\boldsymbol{x})\, ds$

Boundary condition:

$$I_0 = E_0 + \iint k(\boldsymbol{x}, \boldsymbol{n}', \boldsymbol{n}, \nu', \nu)\, I^{(\mathrm{in})}(\boldsymbol{x}, \boldsymbol{n}', \nu')\, d\Omega'\, d\nu'$$

E_0 = specific intensity radiating into the volume
k = surface scattering kernel

References

[1] N. Bathe, A. Tokuta, *Photorealistic Volume Rendering of Media with Directional Scattering*, Third Eurographics Workshop on Rendering, Bristol, UK, May 1992, pp. 227-245.

[2] P. Blasi, B. Le Saec, C. Schlick, *A Rendering Algorithm for Discrete Volume Density Objects*, Computer Graphics Forum, 12(3), 1993, pp. 201-210.

[3] J.F. Blinn, *Light Reflection Functions for Simulation of Clouds and Dusty Surfaces*, Computer Graphics 16(3), 1982, pp. 21-29.

[4] B. Bui-Tuong, *Illumination for Computer-Generated Pictures*, CACM, June 1975, pp. 311-317.

[5] L. Carpenter, *The A-Buffer, an Anti-aliased Hidden Surface Method*, Computer Graphics 18(3), 1984, pp. 103-108.

[6] K.M. Case, P.F. Zweifel, *Linear Transport Theory*, Addison-Wesley, Reading M.A., 1967.

[7] S. Chandrasekhar, *Radiative Transfer*, Oxford University Press, 1950.

[8] B. Corrie, P. Mackerras, *Parallel Volume Rendering and Data Coherence on the Fujitsu AP1000*, Technical Report TR-CS-92-11, Dept. of Computer Science, Australian National University, Nov. 1992.

[9] R.A. Drebin, L. Carpenter, P. Hanrahan, *Volume Rendering*, Computer Graphics 22(4), 1988, pp. 65-74.

[10] D.S. Ebert, R.E. Parent, *Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Dcanmline A-buffer Techniques* Computer Graphics 24(4), 1990, pp. 357-365.

[11] H. Edelsbrunner, *An Acyclicity Theorem in Cell Complexes in d Dimensions*, Proceedings of the ACM Symposium on Computational Geometry, 1989, pp. 145-151.

[12] M.P. Garrity, *Raytracing Irregular Volume Data*, Computer Graphics 24(5), 1990, pp. 35-40.

[13] L.G. Henvey, J.L. Greenstein, *Diffuse reflection in the galaxy*, Astrophys. J. 93, 1941, p. 70.

[14] B.K.P. Horn, R.W. Sjoberg, *Calculating the reflectance map*, Applied Optics 18(11), 1979, pp. 1770-1779.

[15] M. Inakage, *Volume Tracing of Atmospheric Environments*, Visual Computer, 1991, pp. 104-113.

[16] A. Kaufman, E. Simony, *Scan-conversion Algorithms for Voxel-based Graphics*, Proceedings ACM Workshop on Interactive 3D Graphics, Chapel Hill, NC, 1986, pp. 45-75.

[17] A. Kaufman, *Efficient Algorithms for 3D Scan-Conversion of Parametric Curves, Surfaces, and Volumes*, Computer Graphics 21(4), 1987, pp. 171-179.

[18] A. Kaufman, *Efficient Algorithms for 3D Scan-Converting Polygons*, Computers and Graphics, 12(2), 1988, pp. 213-219.

[19] A. Kaufman, *Intermixing Surface and Volume Rendering*, in *3D Imaging in Medicine*, K.H. Höhne, H. Fuchs, S.M. Pizer (Eds.), Springer-Verlag, 1990.

[20] J.T. Kajiya, B.P. Von Herzen, *Ray Tracing Volume Densities*, Computer Graphics 18(3), 1984, pp. 165-174.

[21] J.T. Kajiya, *The Rendering Equation*, Computer Graphics 20(4), 1986, pp. 143-150.

[22] W. Krüger, *Volume Rendering and Data Feature Enhancement*, Computer Graphics 24(5), 1990, pp. 21-26.

[23] W. Krüger, *The Application of Transport Theory to Visualization of 3-D Scalar Data Fields*, Comput. Phys. 5(4), 1991, pp. 397-406.

[24] D. Laur, Pat Hanrahan, *Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering*, Computer Graphics 25(4), 1991, pp. 285-288.

[25] M. Levoy, *Display of Surfaces from Volume Data*, IEEE Computer Graphics and Applications 8(3), 1988, pp. 29-37.

[26] M. Levoy, *A Hybrid Ray Tracer for Rendering Polygon and Volume Data*, IEEE Computer Graphics and Applications 10(2), 1990, pp. 33-40.

[27] W.E. Lorensen, H.E. Cline, *Marching cubes: A High Resolution 3D Surface Construction Algorithm*, Computer Graphics 21(4), 1987, pp. 163-169.

[28] P. Mackerras, *A Fast Parallel Marching-Cubes Implementation on the Fujitsu AP1000*, Technical Report TR-CS-92-10, Dept. of Computer Science, Australian National University, Aug. 1992.

[29] G.W. Meyer, *Wavelength Selection for Synthetic Image Generation*, Computer Graphics & Image Processing, 41, 1988, pp. 57-79.

[30] N. Max, P. Hanrahan, R. Crawfis, *Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions*, Computer Graphics 24(5), 1990, pp. 27-33.

[31] T. Nishita, Y. Miyawaki, E. Nakamae, *A Shading Model for Atmospheric Scattering considering Luminous Distribution of Light Sources*, Computer Graphics 21(4), 1987, pp. 303-310.

[32] K.L. Novins, F.X. Sillion, D.P. Greenberg, *An Efficient Method for Volume Rendering using Perspective Projection*, Computer Graphics 24(5), 1990, pp. 95-103.

[33] H.E. Rushmeier, *Realistic Image Synthesis for Scenes with Radiatively Participating Media*, PhD thesis, Program of Computer Graphics, Cornell University, 1986.

[34] H.E. Rushmeier, K.E. Torrance, *The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium*, Computer Graphics 21(4), 1987, pp. 293-302.

[35] P. Sabella, *A Rendering Algorithm for Visualizing 3D Scalar Fields*, Computer Graphics 22(4), 1988, pp. 51-58.

[36] G. Sakas, J. Hartig, *Interactive Visualization of Large Scalar Voxel Fields*, Proceedings Visualization 1992, Kaufman and Nielsen, Eds., IEEE Computer Science Press, 1992, pp. 29-36.

[37] P. Schröder, W. Krüger, *Dataparallel Volume Rendering Algorithms for Interactive Visualization*, (Preprint) Workshop Visualisierung - Rolle von Interaktivität und Echtzeit, GMD Sankt Augustin, 1992.

[38] P. Schröder, J.B. Salem, *Fast Rotation of Volume Data on Data Parallel Architectures*, Proceedings Visualization 1991, Nielsen and Rosenblum, Eds., IEEE Computer Science Press, 1991, pp. 50-57.

[39] P. Shirley, *Physically Based Lighting Calculations for Computer Graphics*, PhD Thesis, University of Illinois at Urbana-Champaign, 1991.

[40] P. Shirley, A. Tuchman, *A Polygonal Approximation to Direct Scalar Volume Rendering*, Computer Graphics 24(5), 1990, pp. 63-70.

[41] C. Upson and M. Keeler, *V-Buffer: Visible Volume Rendering*, Computer Graphics 22(4), 1988, pp. 59-64.

[42] L. Westover, *Footprint Evaluation for Volume Rendering*, Computer Graphics 24(4), 1990, pp 367-376.

[43] J. Wilhelms, J. Challinger, N. Alper, S. Ramamoorthy, A. Vaziri, *Direct Volume Rendering of Curvilinear Volumes*, Computer Graphics 24(5), 1990, pp. 41-47.

[44] J. Wilhelms, A. Van Gelder, *Topological Considerations in Isosurface Generation*, Computer Graphics 24(5), 1990, pp. 79-86.

[45] J. Wilhelms, A. Van Gelder, *A Coherent Projection Approach for Direct Volume Rendering*, Computer Graphics 25(4), 1991, pp. 275-284.

[46] P.I. Williams, P. Shirley, *An A Priori Depth Ordering Algorithm for Meshed Polyhedra*, Technical Report 1018, Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, September 1990.

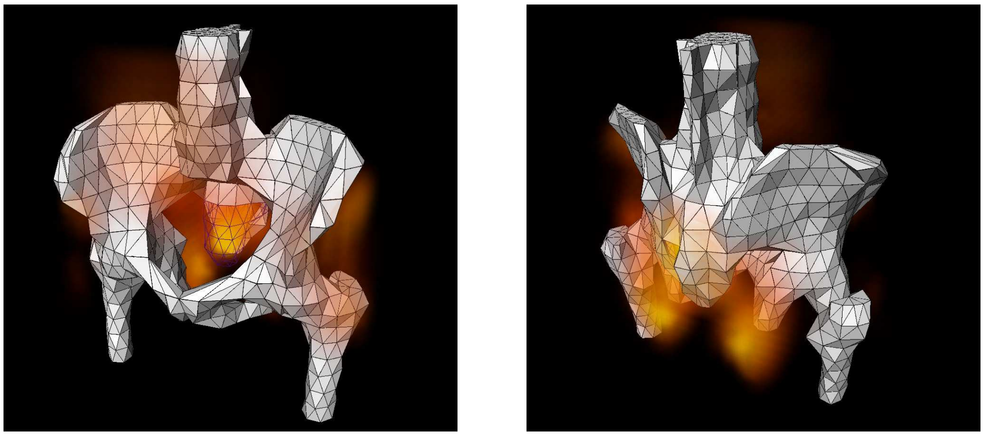


Figure 4: Volume rendering of a three dimensional temperature distribution using a hierarchical splatting technique.

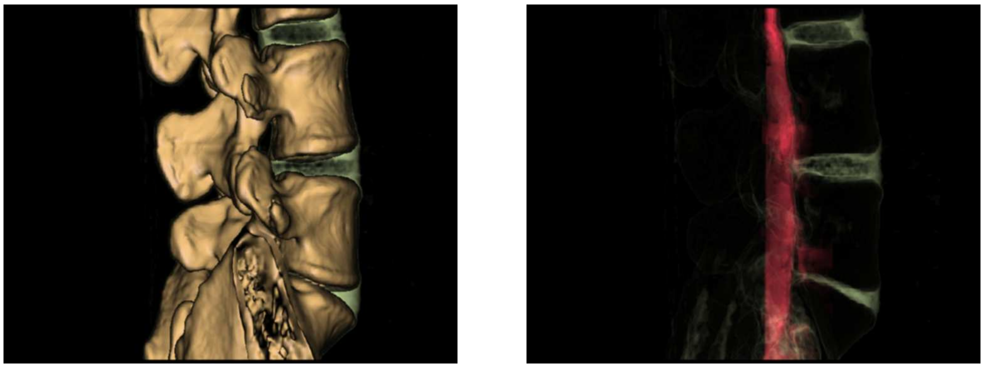


Figure 5: A CT-data set showing the human spinal column visualized using a ray-casting algorithm (by courtesy of W. Krüger).

Figure 6: Room containing a participating medium rendered using a Monte-Carlo based algorithm (by courtesy of P. Shirley).