Light Transfer Equations for Volume Visualization

Francesc Sala i Valcàrcel
Daniela Tost i Pardell

Report LSI-96-58-R

Light Transfer Equations for Volume Visualization.

Francesc Sala i Valcàrcel[†] and Daniela Tost i Pardell[‡]

Departament de Llenguatges i Sistemes Informàtics

Universitat Politècnica de Catalunya

Av. Diagonal 647, 8ª, E–08028 Barcelona

Report LSI–96–58–R

November 1996

Abstract

The simulation of the transport of light through non-empty media is of great importance in volume visualization applications. For low-albedo media, a good approximation to this transport is the Single Scattering Model, which is used extensively. Its derivation is presented here in detail. The model establishes the light that reaches an observer through a participating media from a given direction. The resultant expression is an integral over a line of sight of the light emitted at each point, properly scaled by its correspondent attenuation factor. In the general case, the integral has not closed form solution and must be computed numerically. However, it may be solved analytically if some simplificative assumptions are considered; these assumptions are exposed here, and the corresponding simplified expressions are given. Finally, it is shown how the Single Scattering Model may be applied both to ray-casting and to splatting algorithms, and how it is used to shade surfaces in direct volume rendering.

[†] e–mail : francs@turing.upc.es

[‡] e–mail : dani@turing.upc.es and dani@lsi.upc.es

Contents

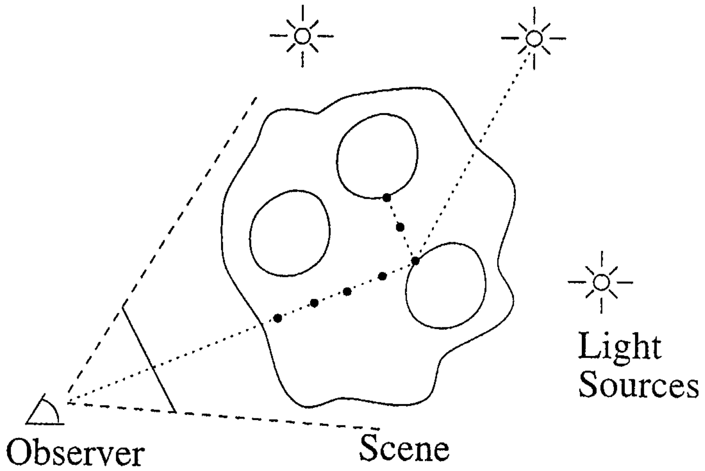


Figure 1: Illumination is computed only on a discrete set of directions and only on a discrete set of points or elements. These points may or not lay on surfaces and may be visible or not from the viewer position.

1 Introduction.

The shading problem in computer graphics consists in computing how much light reaches a virtual observer coming from a scene in a set of given directions.

This is a continuous problem: in order to compute exactly how much light reaches the observer, an infinite number of viewing directions through the scene and infinite points on each direction should be considered. As any continuous problem that has to be solved by a computer, it must be discretized. There are three sources of discretization (see figure 1):

- the number of viewing directions in which light is computed; only one or a few directions for each pixel of the image plane,
- the number of points where the illumination is measured; only a subset of the visible ones, only those laying on surfaces, etcetera,
- and the number of incoming light directions considered at each point; only the directions that are likely to have a significant contribution to the total illumination of the point, typically the ones reaching the point from light sources.

In order to solve the shading problem, it is necessary to model light transport through environments. In general terms, an environment is equivalent to a set of objects inmersed in a participating medium. The goal of this report is to present and to analyze different models of light transport and interaction with both objects surfaces and surrounding media and to relate these models with the existing shading models. The report focuses mainly at light transport through volumes, and it is intended at detecting the open problems in volume rendering.

The report is divided into four main sections. First, the modelling of light is briefly explained in section 2. Next, in section 3, the light transport methods on surfaces are enumerated and some surface shading models are reviewed. The behaviour of light through volumes is presented in section 4. First, volume representation is described (section 4.1), next light transport mechanisms through volumes are presented (section 4.2). Then two derivations of the light transport model are presented in sections 4.3 and 4.4. These derivations are equivalent and lead to the same result. As the explanation of section 4.4 is more direct and easy to follow, for a quick reading section 4.3 may be skiped. The fifth section of the report is devoted to the shading algorithms based on the model of the previous section. Finally, the conclusions are presented in section 6.

2 Light.

In Physics, there have been two approaches in the modelling of light: wave theory and particle theory. The former one models light as an electromagnetic wave that propagates through space in all directions. As such an electromagnetic wave, it may be broken down into perpendicular electric and magnetic fields. This approach, initially due to Huygens and posteriorly to Planck, has given place to the Physical Optics. It allows to explain the effects of light polarization, interference and diffraction.

The latter theory was first outlined by Newton, rejected by Foucault, and finally formalized by Einstein. According to this theory, a light ray is a beam of energy particles, called photons, that travel through space along rectilinear paths. The energy of a photon is a minimum indivisible quantity. The particle theory has given rise to the discipline known as Geometrical Optics. This model allows to explain the photoelectric effect.

These two approaches had been competing one against each other until Louis de Broglie established that any particle that travels at very high speed produces behind it a wave that travels at the same speed and in the same direction. As a consequence, light has a double nature of wave and particles. Therefore, both theories are correct, and complement each other. The discipline that studies at a time the double nature of light is called Quantum Optics.

Up to now, for computer simulations, the particle model has been considered as being more suitable, because the geometrical approach supposes less computational effort. Therefore. the effects of diffraction, interference and polarization are generally not modelled.

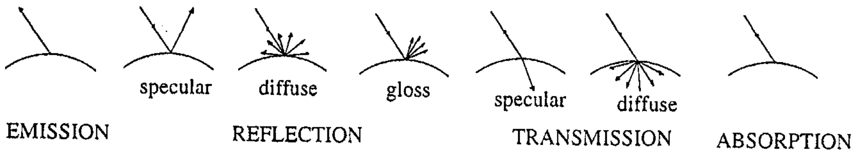


Figure 2: Light Transport Mechanisms.

Accurate explanations on what these behaviours consist in, and experiences that make them clear may be found in [Gla95] and classic literature on Physics and Optics, i.e. [BdE74],[Sea63] and [Com70].

In this report, only the particle model is analyzed.

3 Modelling of light interaction with surfaces.

Although this report mainly addresses volume visualization, in this section the modelling of light interaction with surfaces is described in order to allow a better understanding of the light transport.

The interest is focused on the physical properties of surfaces that determine their behaviour with light and thus the visual perception an observer has of them. The shading models presented herein will be related with a more general transport theory in section 4.3.

3.1 Main light transport mechanisms on surfaces.

The main behaviours of light interaction with surfaces are shown in figure 2.

Emission : Any thermical or chemical process that allows a surface to emit its own light.

Specular Reflection : Propagation of incident light as a perfect mirror, i.e. in a direction such that the angle between the surface normal and the incident and reflected light rays coincide. There is no change in the wavelength of incident light.

Diffuse Reflection : Propagation of incident light in all directions with equal energy.

Glossy Reflection : Propagation of light involving specular and diffuse reflections. Light propagates diffusely around a main direction. Glossy reflection is responsible for a, mirrorlike appearance of a rough surface.

NAME	FORMULAE
Lambert	$I = k_a I_a + k_d \sum_{i=0}^{n-1} I_i (S_i \cdot N)$
Phong	$I = k_a I_a + \sum_{i=0}^{n-1} f_{at_i} I_i \left[k_d (S_i \cdot N) + k_s (R_i \cdot V)^{k_e} \right]$
Blinn-Phong	$I = k_a I_a + \sum_{i=0}^{n-1} f_{at_i} I_i \left[k_d (S_i \cdot N) + k_s (H_i \cdot N)^{k_e} \right]$
Cook-Torrance	$I = R_a I_a + \sum_{i=0}^{n-1} I_i (S_i \cdot N) \cdot (s f_{s_i} + d f_d) \delta \vec{\omega}_i$
Ray tracing	$I = k_a I_a + \sum_{i=0}^{n-1} f_{at_i} g_i I_i \left[k_d (S_i \cdot N) + k_s (H_i \cdot N)^{k_e} \right] + k_s I_s + k_t I_t$
Radiosity	$B_i = E_i + \rho_i \sum_{j=1}^{N_p} B_j F_{ij}$

Table 1: Surface shading models: equations.

Specular Transmission : Propagation of light through a surface, from one medium to another, in a single direction. That direction depends on the angle between the incident ray and the surface normal and the refraction indexes of the two media separated by the surface. The relationship is stated by the Snell's Law.

Diffuse Transmission : Propagation of light through a surface, in all directions with equal energy. When viewing through a surface that presents diffuse transmission, a "blurred-like" vision is perceived.

Absorption : Absorption on a surface occurs when the energy of the incident light is transformed by the surface in any other kind of energy, i.e. heat.

3.2 Shading models on surfaces.

Surface shading models determine the surface's color at a given point on the surface. They rely on factors that express the material's properties, on the orientation of the shaded surface, on the viewer position and its view direction,

NAME	EM.	REFL.			INTERREFL.			TRANSM.
		spec.	diff.	gloss	spec.	diff.	gloss	
Lambert	no	no	yes	no	no	no	no	no
Phong	no	yes	yes	no	no	no	no	no
Blinn-Phong	no	yes	yes	no	no	no	no	no
Cook-Torrance	no	yes	yes	no	no	no	no	no
Ray tracing	no	yes	yes	no	yes	no	no	yes
Radiosity	yes	no	yes	no	no	yes	no	no

EM.	≡	emission	TRANSM.	≡	transmission
REFL.	≡	reflection	spec.	≡	specular
INTERREFL.	≡	interreflection	diff.	≡	diffuse

Table 2: Surface shading models: transmission mechanisms they emulate.

on the light sources locations and on their intensities, and also, in the more realistic models, on the other surfaces in the scene.

In table 1 and table 2, the most known surface shading models are introduced. Table 1 summarizes the equations used in each model; a glossary of terms may be found in appendix A. Table 2 labels the transport mechanisms taken into account in each model. The first four models only account for the light reaching directly from the light sources, and are called *local illumination* models. The last two models also register light coming from the other surfaces in the scene (light coming indirectily from the sources after one or more reflections), and are called *global illumination* models [Sob94]. The table refers to the basic models of ray tracing and radiosity. Advanced models allowing glossy reflections that have recently been published are not included [Chr95].

Lambert model characterizes the diffuse surface reflection that exhibit matte objects. The surfaces of such objects reflect light in all directions with equal intensity. The viewer always perceive them with the same bright, independently of the viewer's position. The amount of diffuse reflection does not depend on the viewer location, but on the orientation of the surface respect to lights. The more perpendicularly a light ray reaches the surface, the more energy receives the surface per unit area, and consequently, the more energy the surface reflects diffusely. As not all materials behave the same under identical illumination conditions, the amount of diffuse reflection also depends on a material coefficient called the diffuse-reflection coefficient, k_d, which varies between 0 and 1. From all the light energy arriving to the surface being shaded, only the one coming directly from light sources is sampled. As a coarse approximation to all the other light energy arriving to the surface which is not sampled, an ambient term is included into the formula. This will also be done in the next models.

Phong model adds specular reflection to Lambert model. Specular reflection is the phenomenon responsible for highlights on shiny surfaces. The maximum specular reflection occurs along the perfectly specular reflection direction, R, and it is considerable along a more or less small solid angle around that direction. So, the viewer perceives a highlight due to specular reflection when the viewing direction V is parallel or almost parallel to R, and this highlight perception decreases rapidly as V moves away from R. Two coefficients, k_s and k_e rule the amount of specular reflection perceived by the viewer. Furthermore, to add realism to the model, an attenuation function is included. This function scales the amount of diffuse and specular reflection, accounting for the fact that light coming from the source is attenuated in its way towards the surface.

The perfectly specular reflection direction R is computed from the lighting direction and the surface normal as $R = 2N(N \cdot S) - S$. A slightly different version of Phong's model, called Blinn-Phong's model, consists of substituting the vector R by the half way vector $H = (S+V)/|S+V|$. It is easier to compute, and if light sources and the viewer are at infinity, it is constant for the entire scene. It produces similar results to the ones of Phong's model.

Unlike the previous methods, which are empirical, the model of Cook and Torrance is physically based. The surfaces of the objects are supposed to be composed by V-shaped grooves, randomly distributed and oriented. The microfacets that form these grooves are planar perfect mirrors. In Cook-Torrance model, specular and diffuse reflections are characterized: light from the source reaching directly the microfacets is responsible for specular reflection, while the interreflections between microfacets cause the diffuse reflection. The formulation is rather complicated, and it is presented in appendix A. The results of Cook-Torrance model differ from those obtained with Phong model in two aspects: the main specular reflection direction and the color of the highlights. In Phong model, the color of the diffuse reflection is the one of the object, while the specular reflection has the color of the light and has its maximum along direction R. In Cook-Torrance model the maximum specular highlight occurs along a direction slightly deviated from R, and the color of the highlight, although mainly set to the light's color, is more or less influenced by the surface's color depending on the light incident angle. Cook-Torrance is more realistic and more time expensive.

To achieve even more realism, the exchange of light between the surfaces in the scene has to be considered. This is done in global illumination models (ray tracing and radiosity). As a consequence, the computation requirements greatly increment.

Impressive images may be computed by ray tracing algorithms. In ray tracing, one or more rays are casted from the viewer's eye through each of the pixels of the image plane to determine visible surface points. For each of those points, the shading model is evaluated. This model consists of the Phong model plus the consideration of light coming from other surfaces along specific directions: the specular reflection of the viewing direction and its transmitted direction according to Snell's law. To compute those contributions, secondary rays are casted along those directions, to reach other surface points where illumination is again calculated. In such a way, the illumination model is recursively computed on surface points, until some termination criterion holds. A geometry factor may easily be introduced to include shadow effects. Ray tracing, then, models specular interreflexions, specular transmissions (transparencies) and shadows. It is an obvious consequence then that it is view dependent.

Ray tracing algorithms work fine for light transfer in specular reflectors environments, but they are not adequate for diffuse reflectors environments, since they do not model the diffuse interreflections. To deal with this latter purpose, radiosity methods were proposed. Surfaces in the environment are all supposed to be perfect diffuse reflectors, and are discretized into patches. These patches are allowed to emit their own light, avoiding thus the point light restriction. Light interactions between patches are determined independently from the viewer's position. The correspondent expression in table 1 represents a set of linear approximated equations which expresses an energy equilibrium in a discretized environment. Once the system is solved, patches may be rendered for any viewer's position.

3.3 Surface characterization respect to light.

As it may be deduced from the previous section, attending to the shading problem, surfaces are modelled as a serie of coefficients or parameters that fall into two main categories:

Geometrical Properties : For instance surface normal, or surface roughness (used to estimate a "deviated normal").

Optical Properties : Such as colour, reflectivity, or transparency, that determine how the surface interacts with light

How many and which of these parameters are used depends on the shading model applied, its accuracy and the effects it tries to emulate. For instance, when applying Phong shading, the surface normal and the specular and the diffuse coefficients are needed. When applying a more sophisticated model, as Cook-Torrance shading, the roughness is also necessary.

The optical coefficients are expected to correspond to the physical reality. It is, the coefficients are supposed to have been obtained by radiometry measures of samples of the material they try to characterize. When this is not possible, the coefficients are set to values that have empirically proved to turn out into satisfactory results.

Some surface shading models have been reviewed succinctely, since their study is not the main goal of this report. More information about surface shading models, about the background assumptions they make, and about reflection and transmission is available in [Hal88], [Gla89] , [FDFH93] , [Gla95] and [Sob94].

4 Modelling of light interaction with volume.

4.1 Volume representation.

4.1.1 What is understood by volume data.

Volume data represent the distribution of some properties through a three dimensional space. In order to allow a better understanding of volume data, three examples will be described: a human bone, a teddy bear, and a gas distribution.

The representation of a human bone obtained by computer tomography shows the bone tissue density distribution. In this case, the volume data set represents an object that have a well defined boundary. But this boundary is not represented. In first place, because the data caption device does not supply a geometric description of the bone surface, but a number of density samples regularly spaced. And secondly because users are not only interested on the bone surface, but also on its interior, which is heterogeneous.

A teddy bear is an object that has also a well defined boundary. But it is a very complex one, almost impossible to represent geometrically. So this object is well suited to be represented as a three dimensional textured volume.

The third example is the spatial distribution of a gas in an environment, a gas cloud. In this case the "object" does not have a well defined boundary, because it is difficult to determine where the cloud begins exactly.

These examples illustrate three different situations: objects that have a well defined boundary, but whose shape and/or location are unknown; objects with a well defined boundary, but too complex to be represented geometrically; and finally, "objects" without a well defined boundary. In all three situations a volume representation is adequate. Here the term "volume" is understood by its most general meaning: volume data represents any non-empty medium without explicit surface representation.

In the next section some volume representations will be briefly enumerated.

4.1.2 Signal reconstruction.

As mencioned in the previous section, the data in volume rendering applications are a finite set of sample points placed in the space $(x_i, y_i, z_i, F(x_i, y_i, z_i))_{i=1..N}$, where F is a continuous function. Depending on the way the points are placed in the space, different kinds of grids are distinguished [1] [Wil91]: *regular grids*,

[1] In all these grids, data is placed at corners of the cells.

that is a set of identical rectangular prisms [2]; *rectilinear grids*, when prisms are not equal; *curvilinear* (or *structured*) *grids*, when the edges of the cells are curved lines; and *unstructured grids*, when the data points are scattered without an implicit connectivity, and are usually grouped at arbitrary polyhedra [3].

The volume representation of a sampled region of space consist of the grid of data points plus a reconstruction function f, allowing to compute an approximation of the value of function F at any point of the region, as F is generally not known.

Volume representations based on regular grids are called *voxel* models [Kau90], and unstructured grids lead to *tetrahedral* models. Hierarchical representations exist, both in three dimensional euclidean space, as for instance *octrees* [BJN$^+$94] or *simplicial complexes* [CFM$^+$94], and in frequency domain, as wavelettes [Mur93].

What must be supposed is that F varies smoothly between sample points, that is, F varies smoothly inside each cell of the grid. If it is not the case, the only solution would be to sample F more frequently to get a finer grid.

The reconstruction methods consist of approximating the unknown F cell by cell. At each cell, F is substituted by the interpolant [4] function f that also varies smoothly, assuming that the error committed in the approximation will be small.

Here, attention will be paid only to interpolation methods for cubical and tetrahedral grids, which are the most common ones; some of the methods may be extended to other grids.

For cubical grids, the simplest interpolation method is the *nearest-point* method or *0-order* method. At each location inside a cell, the value of f is assumed to be the one of the nearest cell corner. With this interpolation method, what is being visualized are voxels: cubes centered at a sample point, inside which the value of F is constant and coincides with the value of the sample. It translates into less image quality.

Another interpolation scheme for cubical grids is *trilinear* interpolation.

$$f(x, y, z) = c_1 + c_2x + c_3y + c_4z + c_5xy + c_6xz + c_7yz + c_8xyz \qquad (1)$$

After substituting in this expression for the eight corners of the cell $(x_i, y_i, z_i, F(x_i, y_i, z_i))_{i=1..8}$, a system of eight linear equations with eight unknowns must be solved. Due to the interpolation requirement in the eight corners, when substituting, $f(x_i, y_i, z_i) = F(x_i, y_i, z_i)$, which is known.

[2] As a particular case of regular grids, cubical grids also exist. A cubical grid is the one whose cells are identical cubes.

[3] As a particular case of unstructured grids, tetrahedral grids also exist. A tetrahedral grid is the one whose cells are tetrahedra, arbitrarily shaped, oriented and scaled.

[4] Interpolant means that in the cells corners -sampled data points- functions f and F must coincide in value.

For tetrahedral cells, the most usual is *linear* interpolation:

$$f(x,y,z) = c_1 + c_2x + c_3y + c_4z \tag{2}$$

The coefficients c_j are again determined from the tetrahedron's four corners $(x_i, y_i, z_i, F(x_i, y_i, z_i))_{i=1..4}$.

In both cases, trilinear interpolation for hexahedral cells and linear interpolation for tetrahedral grids, f is C^0 continuous between cells; that is, positional continuity [5]. The function is continuous between cells, but the gradient is not. This may cause anomalies in the shading.

It is possible, when constructing the interpolation function used inside a cell, to pay attention not only to the corner points of that cell but also to other sample points in the surrounding. In this way, higher order continuity between cells is achieved, at the cost of more computation time to calculate the extra coefficients that appear in the interpolant function.

Nielson et al. in [NFHL91] use an interpolant function that is a sum of sample point values weighted by distance. They also propose to leave aside the interpolation requirement and to calculate a lower degree reconstruction function that does not interpolate the sampled values, but minimizes the quadratic error of the approximation.

The interpolation functions that have been explained vary smoothly inside each cell and, once their expressions have been calculated, are easy to be evaluated at any point. That is why they are so extendly used. But that does not mean that they are the only possible interpolators. It is factible that for some concrete F, other interpolators exist (i.e. not polynomials) that approximate the function closer.

More information about reconstruction functions and further references are provided in [Wil91], [WM92], [NFHL91] and [Max95].

4.2 Light transport mechanisms through volumes.

In this section the behaviour of light through volumes (remember, any nonempty medium) will be explained. To this purpose the conceptual model for volumes is exposed previously.

Conceptually, any volume is intended to be a set of material particles or blobs suspended in a vacuum. The number of these particles is very large, so its distribution may be regarded as a continuum. Moreover, usually the distribution is also considered heterogeneous, that is, the number of material particles is allowed to change from region to region of the volume extent. The density (or whatever the property is sampled) in a point is assumed to be related with the number of particles into a differential volume around the point. Another usual

[5] When estimating the value for a point that belongs to several cells, the same value is obtained with each of the cells interpolants.

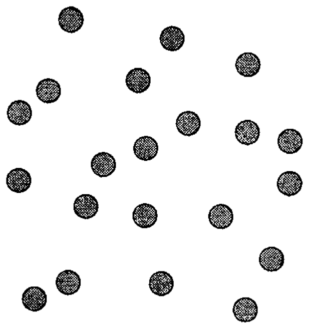


Figure 3: Volume model: a set of numerous material particles suspended in a vacuum.

assumption is to consider spherical and equal sized material particles (figure 3).

These particles may emit photons, absorb and scatter them.

Light transport mechanisms inside a volume are four: emission, streaming, absorption, and scattering. The two latter are caused by a collision between a moving photon and a static material particle.

Emission : This is the name given to any process that injects new photons in the system. Normally, these process have a chemical or a thermical nature.

Streaming : Once emitted, photons travel through volume. It may happen that a photon travel without colliding with any material blob. This behaviour (travelling without suffering any collision) is called streaming. If the refraction effect is neglected, as it is usually done, photons are supposed to travel in straight lines.

Absorption : Occasionally, photons may collide with a material particle. When this occurs it is possible that the material particle absorbs the photon energy, and this one disappears. Then, absorption removes photons from the system. The photon energy is transformed by the material particle into another energy form, as heat for instance.

Scattering : When a collision occurs it may also happen that the photons are not absorbed but deviated from their moving direction to another travelling direction. Usually, photons are supposed not to change their speed when this occurs. The deviation of a photon due to collision is called scattering.

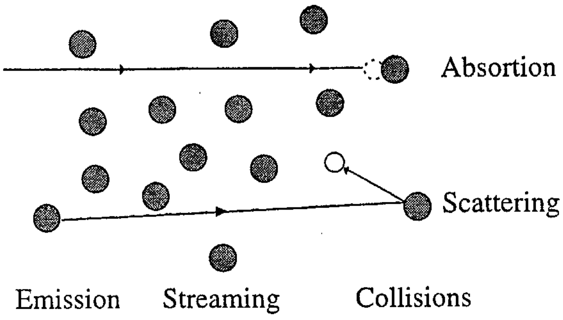


Figure 4: Light transport mechanisms inside a volume.

These are the four mechanisms of light transport inside a volume, and they are ilustrated in figure 4. As it may be seen, transmission is not included among them: the material particles are opaque. Of course, it is possible to include transmission in some special cases, i.e. when pretending to model clouds of water droplets, or ice particles, as it may occur in atmospherical and astronomical applications. But if it is not the case, there is no transmission and particles are totally opaque; this is the case, for instance, in medical volume data sets.

The way a material particle scatters the photons impinging on it depends on its shape, usually supposed to be spherical, and on its size. If the radius of the particle is far smaller than the wavelength of light, no scattering occurs, there is only absorption. If the radius of the particle is similar to the wavelength of light, scattering occurs due to diffraction of light. Finally, if the size of the particle is far much bigger than wavelength of light, the material particle behaves as any other surface, and scattering occurs due to reflection.

4.3 A general equation.

In the present section, a general equation for illumination in scenes composed of volume and surfaces will be derived. The approach of [Arv93] and [Gla95] will be closely followed. First, general moving particles and a complete medium will be considered. Next, the equation will be restricted to photons transport. In section 4.3.2 the formula will be adapted to a simple volume media.

4.3.1 Establishing an equilibrium.

In Computer Graphics, environments are suppoded to be stationary respect to light transfer. Therefore the energy transfer within an environment is in equilibrium state. The transition between one equilibrium to another one is instantaneous and is never modelled.

To solve the rendering problem the photons inside a three dimensional space V (the scene, or part of it) moving in any of a certain set of directions Ω (those

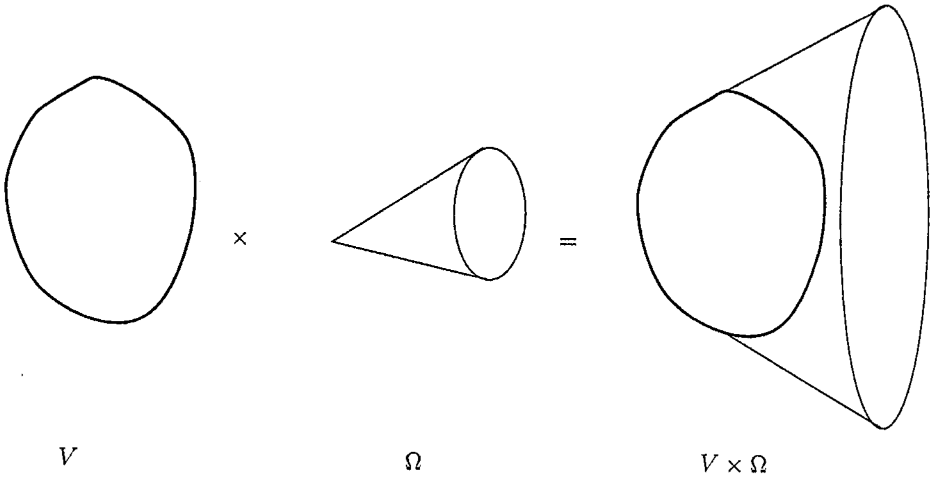


Figure 5: The set $V \times \Omega$ where the interest is focused.

directions that may lead a photon from the scene to the observer's eye) should be counted. Due to equilibrium of energy, this quantity must be a constant.

What follows then, is that all phenomena that may change this quantity must balance each other, in order to make it constant. These phenomena are emission, streaming, absorption and scattering. So,

$$\begin{bmatrix}\text{Changes}\\ \text{due to}\\ \text{Emission}\end{bmatrix} + \begin{bmatrix}\text{Changes}\\ \text{due to}\\ \text{Streaming}\end{bmatrix} + \begin{bmatrix}\text{Changes}\\ \text{due to}\\ \text{Absorption}\end{bmatrix} + \begin{bmatrix}\text{Changes}\\ \text{due to}\\ \text{Scattering}\end{bmatrix} = 0 \quad (3)$$

Next, an analytic expression of the equilibrium in 3 will be derived for an arbitrary volume V and an arbitrary solid angle Ω. Figure 5 shows the set $V \times \Omega$, while the figure 6 depicts graphically all the effects.

As the equilibrium condition stated above must hold for every volume and every solid angle, it must also hold for V and Ω.

If $q(\mathbf{r}, \omega)$ is a function that counts the number of photons emitted at point $\mathbf{r}$ in direction ω per unit time, then the changes in the number of photons in $V \times \Omega$ due to emission are

$$E = \int_\Omega \int_V q(\mathbf{r}, \omega)\ \delta\mathbf{r}\delta\omega \qquad (4)$$

That is, E is the number of photons emitted per unit time inside, V in a direction ω belonging to Ω. Here, ω is a unit vector that represents a direction.

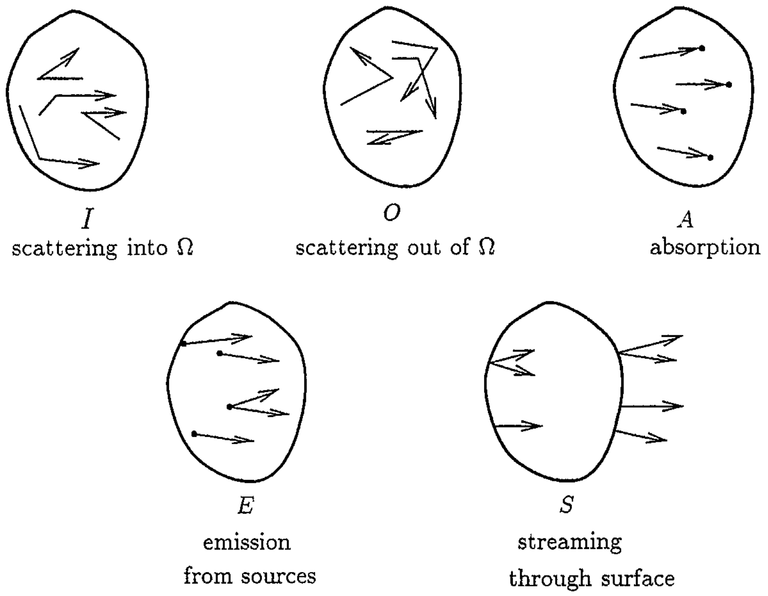


Figure 6: "The five processes affecting the number of particles in volume V with directions in solid angle Ω." (From [Arv93]).

Changes due to streaming are the net flow of photons moving in directions belonging to Ω that pass through the surface of V, δV, per unit of projected area.

$$S = \int_{\Omega} \int_{\delta V} \phi(\mathrm{s}, \omega)\ \omega \cdot n(\mathrm{s})\ \delta \mathrm{s} \delta \omega \tag{5}$$

where $\phi(\mathrm{s}, \omega)$ is the flux of photons at point s moving in direction ω, and $n(\mathrm{s})$ is the normal vector of the surface δV at point s lying on the surface.

As a consequence of Gauss's Theorem (or divergence theorem) and being ∇ the operator

$$\nabla = \mathbf{i}\frac{\delta}{\delta \mathbf{x}} + \mathbf{j}\frac{\delta}{\delta \mathbf{y}} + \mathbf{k}\frac{\delta}{\delta \mathbf{z}} \ , \tag{6}$$

S may be written as:

$$S = \int_{\Omega} \int_{V} \omega \cdot \nabla \phi(\mathbf{r}, \omega)\ \delta \mathbf{r} \delta \omega \tag{7}$$

In such a way that $\omega \cdot \nabla \phi(\mathrm{s}, \omega)$ is the directional derivative of ϕ along direction ω.

Changes in number of photons due to absorption are

$$A = \int_\Omega \int_V \sigma_a(\mathbf{r})\, \phi(\mathbf{r},\omega)\; \delta\mathbf{r}\delta\omega \tag{8}$$

where $\sigma_a(\mathbf{r})$ is the probability, per unit distance, of a photon to be absorbed at point $\mathbf{r}$.

The amount of outscattering is measured as

$$O = \int_\Omega \int_V \int_{S^2} k(\mathbf{r},\omega \cdot \omega')\, \phi(\mathbf{r},\omega)\; \delta\omega'\delta\mathbf{r}\delta\omega \tag{9}$$

where S^2 is the unit sphere, that is the set of all possible directions in space, and $k(\mathbf{r},\omega \cdot \omega')$ is a function that indicates the probability of a photon in $\mathbf{r}$ travelling in direction ω to suffer a collision in $\mathbf{r}$ and be deviated in direction ω'.

In analogous way, inscattering is

$$I = \int_\Omega \int_V \int_{S^2} k(\mathbf{r},\omega' \cdot \omega)\, \phi(\mathbf{r},\omega')\; \delta\omega'\delta\mathbf{r}\delta\omega \tag{10}$$

Then, equation 3 may be rewritten as

$$E - S - A + I - O = 0 \tag{11}$$

Minus signs have been put in streaming, absorption and outscattering, as they are losses.

Or, what is the same,

$$E - A + I - O = S \tag{12}$$

(emission, less absorption, plus inscattering, less outscattering must equal net flux).

$$E + I = S + A + O \tag{13}$$

(gains equal losses). (Note how the graphics in figure 6 accomplish all these equations).

Expanding the five terms and removing outer integrals,

$$q(\mathbf{r},\omega) + \int_{S^2} k(\mathbf{r},\omega' \cdot \omega)\phi(\mathbf{r},\omega')\delta\omega' = \omega \cdot \nabla\phi(\mathbf{r},\omega) + \sigma(\mathbf{r})\phi(\mathbf{r},\omega) \tag{14}$$

being $\sigma(\mathbf{r}) = \sigma_s(\mathbf{r}) + \sigma_a(\mathbf{r})$ the probability that a photon will suffer any kind of collision (turning out into an absorption or turning out into a scattering) per unit distance travelled,

and being $\sigma_s = \int_{S^2} k(\mathbf{r},\omega_0 \cdot \omega')\; \delta\omega'$ the probability of a particle to suffer a scattering collision per unit distance travelled.

Suppressing the outer integrals of E, A, I, O and S means that the equilibrium holds not only for V and Ω, but for any volume and solid angle. In particular, equation 14 makes reference to any differential volume and single direction (differential solid angle).

Even though the explanations made up to now were related to photons, they actually apply to a more general set of moving particles. In fact, equation 14 also rules the transport of neutrons, electrons, gas molecules, and may also be applied to solve a great variety of problems, including fluids dynamics, and traffic, between others. It is known as Boltzman's equation. In general, it applies to situations where a set of moving particles travel through a medium, and interact with it as explained before. The medium is constituted by suspended material blobs that may emit moving particles, absorb and scatter them. The moving particles travel at a constant speed, even after a scattering collision, and do not interact (collide) between them.

When dealing with photons, the constant speed assumption is equivalent to say that all photons have the same frequency. To produce color images, three independent simulations are done and combined, corresponding to the frequencies related to red, green and blue colors.

More than the number of photons moving in an environment, it is the energy carried by them that must be modelled. Therefore, equilibirum is generally expressed in terms of radiance L ("$\cdots$ the power per unit area per unit solid angle due to radiant energy crossing a surface perpendicularly" [Arv93]).

$$\begin{aligned} \varepsilon(\mathbf{r},\omega) + \textstyle\int_{S^2} k(\mathbf{r},\omega' \cdot \omega) L(\mathbf{r},\omega')\,\delta\omega' &= \omega \cdot \nabla L(\mathbf{r},\omega) + \sigma(\mathbf{r}) L(\mathbf{r},\omega) \\ L(\mathbf{s},\omega) &= \varepsilon_b(\mathbf{s},\omega) + \textstyle\int_{H_{\mathbf{s}}^-} k_b(\mathbf{s},\omega' \to \omega) L(\mathbf{s},\omega')\,\delta\omega' \end{aligned} \tag{15}$$

$\varepsilon(\mathbf{r},\omega)$ is the energy associated to the photons emitted at $\mathbf{r}$ in direction ω.

$L(\mathbf{s},\omega)$ is the radiance, energy associated to the flux at point $\mathbf{r}$ in direction ω, $\phi(\mathbf{r},\omega)$.

The first equation of 15 is an integro-differential equation (the unknown L appears derived ($\nabla L(\mathbf{r},\omega)$) and integrated ($\int_{S^2}$)). The boundary condition used in order to solve equation 15 sets the value of radiance where the volume ends, that is, on a surface. This boundary condition is the second equation in 15. The point s is a point on a surface, and $H_{\mathbf{s}}^-$ is the set of all incoming directions to s from the volume.

An equivalent, but more compact, condition on L is obtained by integrating equation 15 along a direction ray until a boundary point is reached:

$$L(\mathbf{r},\omega) = \beta(\mathbf{s},\mathbf{r})\mathbf{\Gamma}(\mathbf{s},\omega) + \int_0^h \beta(\mathbf{r} - xw,\mathbf{r})\mathbf{Q}(\mathbf{r} - xw,\omega)\,\delta x \quad , \tag{16}$$

where,

h is the distance from $\mathbf{r}$ to $\mathbf{s}$ (in such a way, $\mathbf{r} - hw = \mathbf{s}$),

$\beta(\mathbf{r}, \mathbf{r}') = e^{-\tau(\mathbf{r},\mathbf{r}')}$ is called path absorption and is the attenuation of light in the path from $\mathbf{r}$ to $\mathbf{r}'$ due to absorption and outscattering,

$\tau(\mathbf{r}, \mathbf{r}') = \int_0^{\|\mathbf{r}'-\mathbf{r}\|} \sigma(\mathbf{r} + y\omega)\, \delta y$ is called optical distance, and $\mathbf{r}$, $\mathbf{r}'$ lie on a ray with direction ω,

$\mathbf{Q}(\mathbf{r}, \omega) = \varepsilon(\mathbf{r}, \omega) + \int_{S^2} k(\mathbf{r}, \omega' \cdot \omega) L(\mathbf{r}, \omega')\, \delta\omega'$ is the volumetric emission at point $\mathbf{r}$ as an addition of volumetric self-emission and inscattering,

and $\mathbf{\Gamma}(\mathbf{s}, \omega) = \varepsilon_b(\mathbf{s}, \omega) + \int_{H_s^-} k_b(\mathbf{s}, \omega' \to \omega) L(\mathbf{s}, \omega)\, \delta\omega'$ is the surface emission at point $\mathbf{s}$ on a surface, as an addition of surface self-emission and surface in-scattering. $\mathbf{s}$ is the point where a ray departing from $\mathbf{r}$ in direction $-\omega$ first encounters a surface.

As equation 15, equation 16 is not a closed form for $L(\mathbf{r}, \omega)$, because $\mathbf{Q}$ and $\mathbf{\Gamma}$ also depend on L. Equation 16 is a condition that radiance must satisfy.

What equation 16 means may be easily understood with the help of figure 7. Point $\mathbf{r}$ in space may be any point, but it will be more interesting if it is the point where the observer is placed. To calculate the light energy reaching $\mathbf{r}$ coming from direction ω, first the point $\mathbf{s}$ is determined. $\mathbf{s}$ is the point where a ray in direction $-\omega$ starting in $\mathbf{r}$ first intersects a surface. So, it is a visible surface point from $\mathbf{r}$. Light reaching $\mathbf{r}$ in direction ω is the one leaving $\mathbf{s}$ in direction ω, that is $\mathbf{\Gamma}(\mathbf{s}, \omega)$. However, in its way from $\mathbf{s}$ to $\mathbf{r}$, light $\mathbf{\Gamma}(\mathbf{s}, \omega)$ is attenuated due to absorption and outscatter (not all photons leaving $\mathbf{s}$ will get $\mathbf{r}$). The portion of $\mathbf{\Gamma}(\mathbf{s}, \omega)$ that reach $\mathbf{r}$ is $\beta(\mathbf{s}, \mathbf{r})\mathbf{\Gamma}(\mathbf{s}, \omega)$.

Furthermore, consider any point $\mathbf{r}'$ between $\mathbf{r}$ and $\mathbf{s}$, also on the ray. $\mathbf{r}'$ is a point on volume, and as such a point it may emit photons towards $\mathbf{r}$ due to volumetric self-emission and/or inscattering, $\mathbf{Q}(\mathbf{r}, \omega)$. And again, not all this energy will reach $\mathbf{r}$, because of absorption and outscattering in the way from $\mathbf{r}'$ to $\mathbf{r}$, making $\beta(\mathbf{r}', \mathbf{r})\mathbf{Q}(\mathbf{r}', \omega)$ the contribution of $\mathbf{r}'$ to the energy at $\mathbf{r}$. To add up all the contributions of intermediate points on the path from $\mathbf{r}$ to $\mathbf{s}$, an integration along that path must be performed.

4.3.2 Simplifications to the general equation.

The model of equation 16 characterizes the light transfer through a complete medium able to emit, scatter and absorb photons. In this way, it is well suited to applications that simulate participating media. But its use is not restricted to such applications. In fact, most of the shading models used in computer graphics, global or local, through a participating medium or through a vacuum, are subsumed in the model of equation 16. It is just a question of terms simplification, usually related to consider a more simple medium. The simplifications more relevant to computer graphics are next presented.

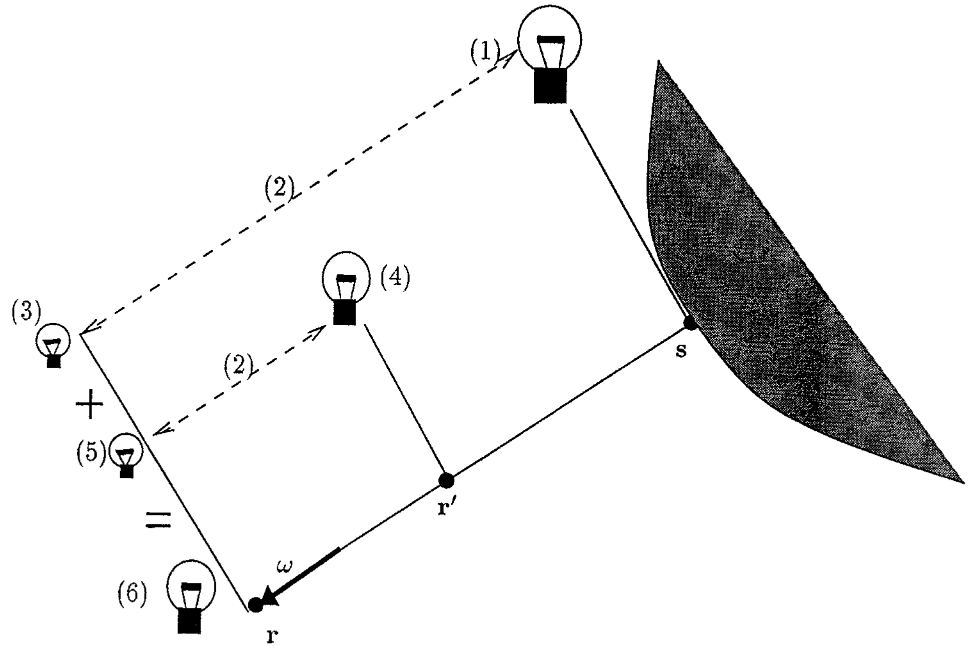


Figure 7: Light reaching r is a combination of light departing from s and contributions along the path from r to s, all of them properly attenuated because of absorption and outscattering. (1) $\Gamma(\mathrm{s},\omega)$, surface emission and surface inscattering. (2) volume absorption and volume outscattering. (3) $\beta(\mathrm{s},\mathbf{r})\Gamma(\mathrm{s},\omega)$, light from surface after attenuation. (4) $\mathbf{Q}(\mathrm{r}',\omega)$, volume emission and volume inscattering, at $\mathbf{r}'$. (5) $\beta(\mathrm{r}',\mathrm{r})\mathbf{Q}(\mathrm{r}',\omega)$ light from r' after attenuation. (6) $L(\mathbf{r},\omega)=\beta(\mathrm{s},\mathrm{r})\Gamma(\mathrm{s},\omega)+\int_0^h\beta(\mathrm{r}-x w,\mathrm{r})\mathbf{Q}(\mathrm{r}-x w,\omega)\ \delta x$, total amount of light reaching r.

4.3.2.1 Light transfer through vacuum.

Under vacuum conditions, no volume emission nor scattering and nor absorption exist, as the medium does not have suspended particles.

Consequently,

$$\begin{array}{ll} \mathbf{Q}(\mathbf{r},\omega) = 0 & \forall\mathbf{r}\forall\omega \\ \sigma(\mathbf{r}) = 0 & \forall\mathbf{r} \\ \beta(\mathrm{s},\mathbf{r}) = 1 & \forall\mathrm{s}\forall\mathbf{r} \end{array}$$

and thus, $L(\mathbf{r},\omega) = \mathbf{\Gamma}(\mathrm{s},\omega)$.

That is,

$$L(\mathbf{r},\omega) = \epsilon_b(\mathrm{s},\omega) + \int_{H_\mathrm{s}^-} k_b(\mathrm{s},\omega' \rightarrow \omega)\, L(\mathrm{s},\omega')\, \delta\omega' \qquad (17)$$

This is the Kajiya's rendering equation [Kaj87]. $L(\mathbf{r},\omega)$ is the light arriving to $\mathbf{r}$ from direction ω. s is the surface point first encountered when travelling from $\mathbf{r}$ along direction $-\omega$. The set H_s^- is the set of all incoming directions of s. $L(\mathrm{s},\omega')$ is the light energy arriving to s from direction ω'. $k_b(\mathrm{s},\omega \rightarrow \omega')$ is the bidirectional reflection distribution function at s, which expresses the ratio of light arriving to s from direction ω' that is reradiated along direction ω.

All the surface shading models explained in section 3.2 are coarse approximations to the solution of the equation. They approximate the exact solution both by considering a simple expression for $k_b(\mathrm{s},\omega \rightarrow \omega')$ and by considering only a discrete set of directions in H_s^-. Local illumination models consider only the directions joining light sources to s, approximating the rest of the integral by an ambient reflection. Ray tracing further consider specular reflection and transmission directions. And radiosity consider a wider, but still small, set of directions: one or more from each patch to s. For Lambert and radiosity, $k_b(\mathrm{s},\omega' \rightarrow \omega)$ is set to $\omega' \cdot \omega$, that is, the cosine of the angle between ω' and ω. (See figure 8 for transfer mechanisms accounted in surface shading models).

Kajiya proposes to solve the rendering equation more accurately via Monte Carlo. Other statistical solving techniques may be found in [AK90].

4.3.2.2 Light transfer through a non-scattering media.

This is the most relevant simplification to volume rendering, and as a consequence, it will be of main importance here.

Consider a medium that emits and absorbs photons but does not scatter them. That is,

$$\begin{array}{ll} \sigma_s(\mathbf{r}) = 0 & \forall\mathbf{r} \\ \mathbf{Q}(\mathbf{r},\omega) = \epsilon(\mathbf{r},\omega) & \forall\mathbf{r}\forall w \end{array}$$

Then equation 16 is simplified to

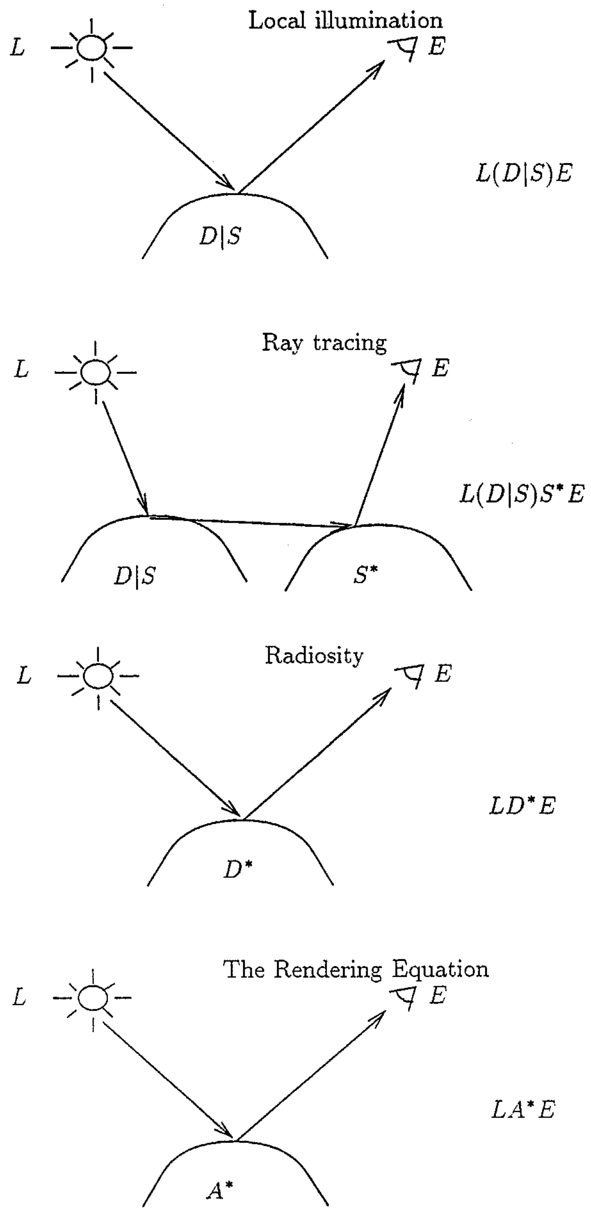


Figure 8: Key: D Diffuse Surface. S Specular Surface. A Arbitrary Surface. L Light Source. E Eye point. * 0 or more. "Possible light paths for local illumination, ray tracing, radiosity, and the Rendering Equation". (From [Sob95]).

$$L(\mathbf{r},\omega) = \beta(\mathbf{s},\mathbf{r})\Gamma(\mathbf{s},\omega) + \int_0^h \beta(\mathbf{r} - xw, \mathbf{r})\epsilon(\mathbf{r} - xw, \omega)\,\delta x \tag{18}$$

In volume rendering it is usual to deal with pure volume; that is, not opaque nor semitransparent surfaces are present, at less explicitly, in the volume representation. In such a case, the integration along direction ω from $\mathbf{r}$ will not end at a surface point; it should end at the point where volume ends. Define $\mathbf{r_e}$ as the last volume point encountered when travelling from $\mathbf{r}$ along direction $-\omega$. If D is the distance from $\mathbf{r}$ to $\mathbf{r_e}$, then $\mathbf{r_e} = \mathbf{r} - D\omega$. And if the volume extent is placed over a background with intensity ϵ_{bkg}, this intensity should play the role of $\Gamma(\mathbf{s},\omega)$. Equation 18 must then be rewritten as

$$L(\mathbf{r},\omega) = \beta(\mathbf{s},\mathbf{r})\epsilon_{bkg} + \int_0^D \beta(\mathbf{r} - xw, \mathbf{r})\epsilon(\mathbf{r} - xw, \omega)\,\delta x \tag{19}$$

Substituting β in equation 19 by its expression (see comments to equation 16) yields

$$L(\mathbf{r},\omega) = e^{-\int_0^D \sigma(\mathbf{r}+xw)\delta x}\epsilon_{bkg} + \int_0^D e^{-\int_0^x \sigma(\mathbf{r}+yw)\delta y}\epsilon(\mathbf{r} - xw, \omega)\,\delta x \tag{20}$$

The meaning of this equation should be clear from its derivation. Anyway, it will be put on evidence later, in section 4.4.5.

4.4 Lighting equations for volume rendering.

The approach followed in section 4.3 is elegant because it relates the light transport equation used in volume visualization to the more complex light transport equations and to the transport equation of moving particles in general. In this way, this approach establishes a binding between radiative transfer and global illumination. The drawback of this approach is that it may result tedious and difficult to follow.

Instead of considering a medium that emits, scatters and absorbs, it is also possible to consider since the beginning a medium that only emits and absorbs, but does not scatter. Such an approach is followed in [Max95]. Max's exposition is short and easy to understand, and it will be followed in this section.

4.4.1 Absorption and Emission.

The considered medium is composed of a large number of particles able to emit and to absorb photons, but not to scatter them. The material particles are suspended in a vacuum, and are supposed to be spherical and equal sized.

Consider now a small cylindrical slab around the viewing direction and perpendicular to it as shown in figure 9. The slab has a base area equal to S, and a width equal to Δt. If ρ is the number of suspended particles per unit volume,

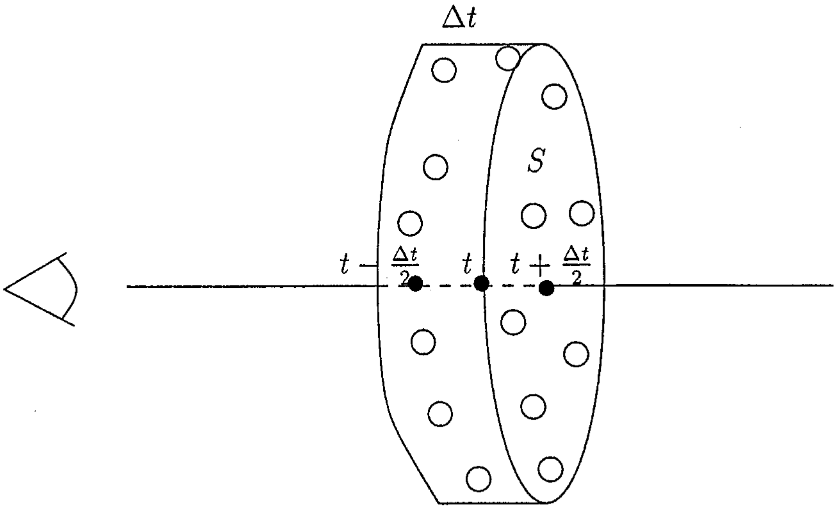


Figure 9: A small cylindrical slab perpendicular to the viewing direction, centered at point t.

then $\rho S\Delta t$ is the total number of particles inside the slab. Being r the radius of a particle, the projection of a particle on to the slab base plane is πr^2. If Δt is small enough, the projections of the particles do not overlap, and then the projection of all the particles inside the slab on its base plane results in an area equal to $\rho S\Delta t\pi r^2$. This projected area is important because it influences the amount of light perceived from the viewer position. The larger this area is, the more light coming from behind the slab is occluded inside it. On the other hand, this area is also the one the particles show to the viewer. Thus, the larger this area is, the more light emitted inside the slab is perceived by the viewer.

Light coming from behind the slab going towards the observer that reaches $t+\frac{\Delta t}{2}$ has a probability of $\frac{\rho S\Delta t\Pi r^2}{S} = \rho\Delta t\Pi r^2$ to be occluded by a particle inside the slab and not to reach $t-\frac{\Delta t}{2}$. If C is the radiance contribution of a particle per unit projected area, then $C\rho\Delta t\Pi r^2$ is the contribution per unit projected area of all the particles inside the slab. Consequently,

$$I(t-\frac{\Delta t}{2}) = I(t+\frac{\Delta t}{2}) + C\rho(t)\Pi r^2\Delta t - \rho(t)\Pi r^2 I(t+\frac{\Delta t}{2})\Delta t \qquad (21)$$

$$I(t-\frac{\Delta t}{2}) - I(t+\frac{\Delta t}{2}) = C\rho(t)\Pi r^2\Delta t - \rho(t)\Pi r^2 I(t+\frac{\Delta t}{2})\Delta t \qquad (22)$$

When moving from $t+\frac{\Delta t}{2}$ to $t-\frac{\Delta t}{2}$, the first term in equation 22 is the gain in light energy due to emission inside the slab; while the second term is the loss in light energy due to absorption inside the slab.

$$I(t+\frac{\Delta t}{2}) - I(t-\frac{\Delta t}{2}) = -C\rho(t)\Pi r^2\Delta t + \rho(t)\Pi r^2 I(t+\frac{\Delta t}{2})\Delta t \tag{23}$$

Moving along the opposite direction, from $t-\frac{\Delta t}{2}$ to $t+\frac{\Delta t}{2}$, what previously was a gain now becomes a loss, and inversely. Light emitted inside the slab, which would be perceived at $t-\frac{\Delta t}{2}$ is not perceived at $t+\frac{\Delta t}{2}$. Light that will be occluded between $t+\frac{\Delta t}{2}$ and $t-\frac{\Delta t}{2}$ si not yet occluded at $t+\frac{\Delta t}{2}$ and may be perceived from that point.

Calling

$$\Delta I = I(t+\frac{\Delta t}{2}) - I(t-\frac{\Delta t}{2}), \tag{24}$$

$$\tau(t) = \rho(t)\Pi r^2, \tag{25}$$

and

$$\epsilon(t) = C\tau(t), \tag{26}$$

the equation 23 now transforms into

$$\Delta I = -\epsilon(t)\Delta t + \tau(t)I(t+\frac{\Delta t}{2})\Delta(t) \tag{27}$$

$$\frac{\Delta I}{\Delta t} = -\epsilon(t) + \tau(t)I(t+\frac{\Delta t}{2}) \tag{28}$$

When Δt approximates to 0, the increments become differentials:

$$\frac{\delta I}{\delta t} = -\epsilon(t) + \tau(t)I(t) \tag{29}$$

$$\frac{\delta I}{\delta t} - \tau(t)I(t) = -\epsilon(t) \tag{30}$$

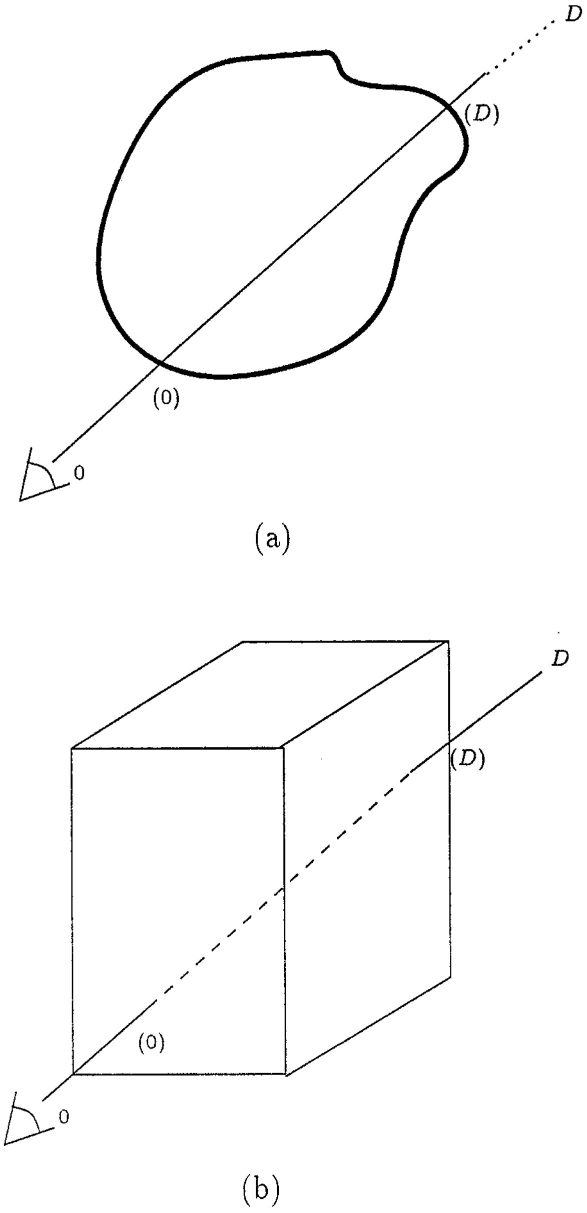
The integrand factor of this ODE is $\exp\left(-\int_0^t \tau(u)\delta u\right)$ (see appendix B). Multiplying both sides of equation 30 by its integrand factor,

$$\left(\frac{\delta I}{\delta t} - \tau(t)I(t)\right)\cdot\exp\left(-\int_0^t \tau(u)\delta u\right) = -\epsilon(t)\cdot\exp\left(-\int_0^t \tau(u)\delta u\right) \tag{31}$$

$$\frac{\delta}{\delta t}\left(I(t)\cdot\exp\left(-\int_0^t \tau(u)\delta u\right)\right) = -\epsilon(t)\cdot\exp\left(-\int_0^t \tau(u)\delta u\right) \tag{32}$$

Now, integrating both sides from 0 to D,

$$I(D)\cdot\exp\left(-\int_0^D \tau(u)\delta u\right) - I(0) = \int_0^D -\epsilon(t)\cdot\exp\left(-\int_0^t \tau(u)\delta u\right)\delta t \tag{33}$$



(a)

(b)

Figure 10: (a) The extrems of integration: $t = 0$ is the viewer position or alternatively the point where the ray enters the volume; $t = D$ is the infinity, or alternatively where the ray intersects an opaque surface, or alternatively where the ray exits the volume. (b) Usually the volume is represented as a three-dimensional array of cubic cells; in that case the extrems of integration are the points where the ray enters and exits the array.

$$I(0) = I(D) \cdot \exp\left(-\int_0^D \tau(u)\delta u\right) + \int_0^D \epsilon(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t \qquad (34)$$

The point $t = 0$ corresponds to the viewer position, or alternatively, to the point where the viewing rays enter the volume extent, while $t = D$ corresponds to the point where the viewing ray exits the volume extent. This is shown in figure 10.

The point $t = D$ is the point where the volume ends. If the scene being rendered is a volume surrounded by surfaces, then $t = D$ corresponds to the point where the viewing ray first intersects one of the surfaces. Therefore $I(D)$ is the light intensity coming from that point on the intersected surface. Frequently, in volume visualization applications, the scene is only composed of pure volume. In that case, $I(D)$ is just a constant background intensity.

Next, $I(0)$ is simply substituted by I, and $I(D)$ by I_b. This term I_b is the intensity coming from behind the volume, either coming from a surface or being a constant background intensity.

$$I = \int_0^D \epsilon(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t + I_b \cdot \exp\left(-\int_0^D \tau(t)\delta t\right) \qquad (35)$$

It should be noted that some authors neglect this second term in equation 35, which is equivalent to consider $I_b = 0$: a black background. Anyway, whatever visualization algorithm is used, the term $I_b \cdot \exp\left(-\int_0^D \tau(t)\delta t\right)$ is easy to be added.

The meaning of the terms included in equation 35 is next described in more depth.

4.4.2 Explanation of some terms.

The term $\tau(t)$ receives several names as extinction coefficient, optical density, interaction coefficient and also, abusively, opacity. $\tau(t) = \rho(t)\Pi r^2$, where $\rho(t)$ is the density at point t (defined as the number of particles per unit volume) and Πr^2, the projected area of a particle (if particles are considered to be spheres with radius r). As the particle model is an abstraction that does not always correspond to the reality (i.e., brain is not composed of spherical particles), the expression $\tau(t) = \rho(t)\Pi r^2$ is not always computable. Instead of that, what is done in practice is to map τ to the scalar field f being visualized. This mapping is done by means of, what is called, a transfer function. $\tau(t) =$ `Transfer_function`($f(t)$), for some `Transfer_function`. This expression is much more useful.

The function $\tau(t)$ varies from 0 to $+\infty$. If normalized, $\hat{\tau}(t) = 1 - e^{-\tau(t)}$ varies from 0 to 1 [WM92] and it is understood as the probability of light incident at t towards the observer to be occluded at t.

The source term $\epsilon(t)$ represents the light intensity in the view direction contributed by the particles inside a differential volume around t, due to self emission. Considering particles, as it has been done, leads to the expression $\epsilon(t) = C(t)\tau(t)$ (equation 26). The expression $C(t)$ may be a constant [Sab88][MHC90], or may effectively depend on t. Later on, $C(t)$ has been set to be a function also depending on the wavelength of light [WM92]. But for the expression of $\epsilon(t)$, it is also possible not to rely on particles and to consider the functions $\epsilon(t)$ and $\tau(t)$ simply continuous functions that characterize the optical properties of the medium. This latter approach is more flexible, because is not restricted to $\epsilon(t) = C(t)\tau(t)$, allowing $\epsilon(t)$ to be mapped independently from $\tau(t)$ [WG91] [CCH95]. As Nelson Max points out in [Max95], furthermore being possible to define ϵ in such a way that it is not related to τ, it is even possible to map them from different scalar fields.

The expression $\exp\left(-\int_0^t \tau(u)\delta u\right)$ is the transparency of the medium between point t and the observer. By analogy, $\exp\left(-\int_0^D \tau(t)\delta t\right)$ is the transparency of the entire volume in the viewing direction, parametrized by t; and $T_i = \exp\left(-\int_{t_i}^{t_{i+1}} \tau(u)\delta u\right)$ is the transparency of the segment from t_i to t_{i+1}. The expression $\alpha_i = 1 - T_i = 1 - \exp\left(-\int_{t_i}^{t_{i+1}} \tau(u)\delta u\right)$ is the opacity of the segment. It is possible to work with transfer functions that directly map the scalar field to transparencies and opacities.

The optical properties of a medium, that is τ, ϵ, T, α are all functions that strongly depend on the wavelengthof light. For color images, they are approximated as a vector of three components, R, G and B. So, the values of these functions are set to scalar values for gray-level images and are set to vector values for color images. As mentioned above, their values are computed using transfer functions, which are often greatly arbitrary.

4.4.3 Adding a Single Scattering.

The optical model explained in the previous subsection models only self-emission and absorption and is mathematically expressed in equation 35. It may easily be extended, replacing the source term $\epsilon(t)$ by a new source term $g(t)$ that not only accounts for self-emission, but also allows a single scattering.

$$I = \int_0^D g(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t + I_b \cdot \exp\left(-\int_0^D \tau(t)\delta t\right) \qquad (36)$$

being $g(t)$ the addition of the self-emission term $\epsilon(t)$ and a new scattering term $S(t)$:

$$g(t) = \epsilon(t) + S(t) \qquad (37)$$

There are two kinds of scattering: volume scattering and surface scattering.

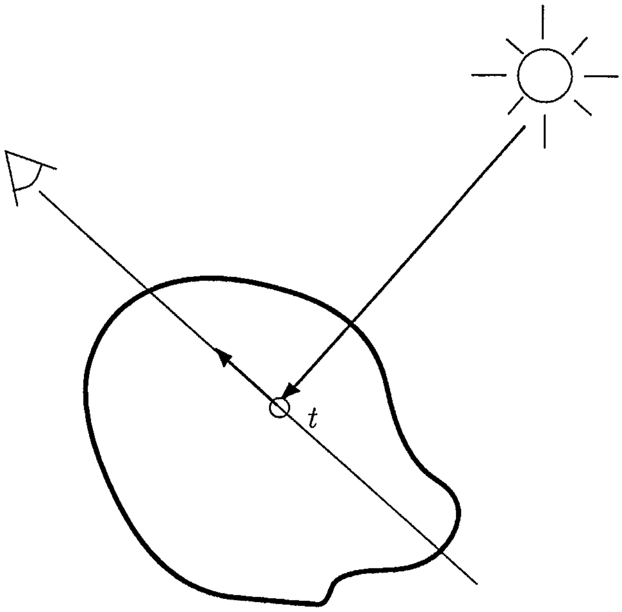


Figure 11: An in-scattering event at t of the light coming directly from the source.

Volume scattering

Equation 35 was an exact expression of the light reaching the observer for a medium that could emit and absorb light, but did not scatter it. Equation 36 is an aproximation of the light reaching the observer through a medium that emits, absorbs and scatters light.

The scattering term $S(t)$ in equation 37 counts the amount of light coming from the light sources that is scattered by the particles at t in the view direction, that is, an in-scattering event (figure 11).

As the particles of the medium scatter light, a great amount of interaction between particles may exist. The contribution of light incorporated to the view direction after several scattering collisions (figure 12) should also be counted.

The phenomenon depicted in figure 11 is called single scattering. Multiple scattering may also appear when several collisions are considered, as it is shown in figure 12. The optical model of equation 36 only considers the single scattering. The albedo of a particle is the ratio of its incident light that is scattered in any direction; that is, the albedo is the particle's reflectivity. If the albedo is low, the contribution of multiple scattering is neglectable. So, equation 36 is a good approximation in the low albedo case.

The volume scattering at a position X in a direction ω is characterized as

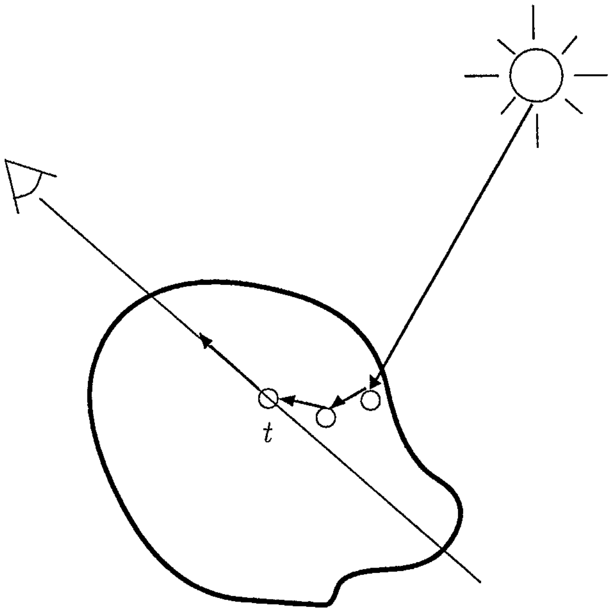


Figure 12: A multiple scattering path.

$$S(X,\omega) = r_v(X,\omega',\omega)i(X,\omega') \tag{38}$$

$$r_v(X,\omega',\omega) = a(X)\tau(X)p(\omega',\omega) \tag{39}$$

ω and ω' could be any direction, but in this formulation they must be set to the viewing direction and the lighting direction, respectively,

$i(X,\omega')$ is the light intensity reaching X coming in direction ω', that is, from the light source,

$r_v(X,\omega,\omega')$ is the bidirectional reflection distribution function,

$a(X)$ is the albedo of the particles at X,

and $p(\omega',\omega)$ is called phase function and specifies the directionality of the scattering.

Although equation 38 manages only one external light source, it may easily be generalized to several external light sources. At the beginning of this section, scattering wasn't considered, so the ratio of light occlussion τ in equation 35 was only due to absorption. Now in fact this occlussion is due both to abşorption and scattering, so $\tau(t) = \tau_a(t) + \tau_s(t)$, being $\tau_s(t) = a(t) + \tau(t)$ and $\tau_a(t) = (1 - a(t))\tau(t)$.

Some volume phase functions are discussed in [Bli82] and in [Gla95].

Surface scattering

Sometimes in volume rendering the user's interest is put on the visualization of isosurfaces. For instance, in medical imaging, it is important to visualize the surface of the organs. In such circumstances, when the goal is to enhance a surface within the volume, surface scattering is used. The surface scattering term is expressed again as

$$S(X,\omega) = r_s(X,\omega,\omega')i(X,\omega') \tag{40}$$

but now, the bidirectional reflection distribution function $r_s(X,\omega,\omega')$ corresponds to the ones for surfaces, as the ones used in surface shading models, i.e., Lambert or Phong. This function requires the normal vector of the surface, which is unknown, but may be estimated in several ways, as it is described in section 5.3.2. Usually, it is aproximated at each point X as the gradient vector of the field f that is visualized.

The model of equation 36 is the most used in volume visualization. It is intended to simulate the propagation of light through a low albedo medium. It allows self-emission, absorption, and a single scattering of the light coming directly from the source. In the next it will be called Single Scattering Model.

4.4.4 Shadowing and Multiple Scattering.

Light coming from the source that is reflected towards the observer by a particle or by a surface, suffers from attenuation along all its path. That is, not only the light is attenuated from the particle/surface to the viewer, but also from the light source to the particle/surface. These effects are called blocking or shadowing and masking, respectively (see figure 13).

Wether shadowing is considered or not depends on which expression is used for $i(X,\omega')$ in equation 38 and 40. If L is the light intensity of the source and $i(X,\omega'$ is set to $i(X,\omega') = L$, shadowing is not considered. To consider the shadowing effect, $i(X,\omega')$ must be set to

$$i(X,\omega') = L \cdot \exp\left(-\int_0^k \tau(LP + u'\omega')\delta u'\right) \tag{41}$$

being LP the lihgt position, and k the distance from the light source to X.

As it has already been explained, multiple scattering accounts for the interreflections between particles. Their consideration implies the solving of integral equations, usually by means of stochastic methods. In [Max95] an abstract of solving strategies is made.

Although shadowing and multiple scattering are supposed to produce a more realistic rendering, they complicate the equations, and the time to compute

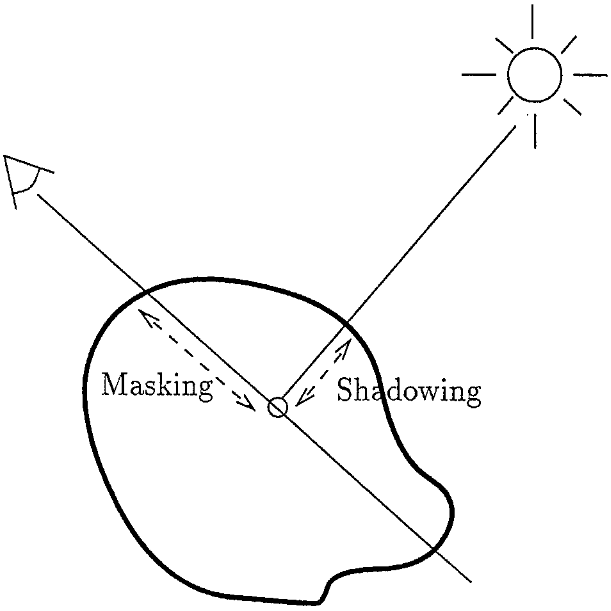


Figure 13: Light coming from the source suffers from attenuation along all its path towards the viewer: shadowing and masking.

them increase greatly. It is not clear whether or not this extra computation effort is justified. In applications that model real environments inmersed into a participating media, these effects are considered, because the goal is to produce a realistic image. But in medical and scientific visualization applications, result images are not supposed to be realistic. Most papers do not take into account shadowing nor multiple scattering. So, attention will be payed only to the Single Scattering Model without shadowing.

4.4.5 Equivalence of approaches.

In section 4.3, the derivation by Arvo in [Arv93] was followed. A complete medium was first considered, and later on simplified to a medium that only absorbed and emmitted light. In Max's paper [Max95], whose derivation has been followed in section 4.4, such a simple medium was considered since the beginning. The resultant equations of such approaches, equations 20 and 35 respectively, are equivalent, as it will be showed here.

Remember both equations:

$$L(\mathbf{r},\omega) = e^{-\int_0^D \sigma(\mathbf{r}+xw)dx} \epsilon_{bkg} + \int_0^D e^{-\int_0^x \sigma(\mathbf{r}+yw)dy} \epsilon(\mathbf{r}-xw,\omega)\, dx \tag{42}$$

$$I = \int_0^D \epsilon(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t + I_b \cdot \exp\left(-\int_0^D \tau(t)\delta t\right) \tag{43}$$

Defining the functions

$$\begin{aligned} \hat{L}(x) &= L(\mathbf{r}-xw), \\ \hat{\sigma}(x) &= \sigma(\mathbf{r}-xw), \\ \text{and} \qquad \hat{\epsilon}(x) &= \epsilon(\mathbf{r}-xw), \end{aligned}$$

equation 42 rewrites as [6]

$$\hat{L}(0) = e^{-\int_0^D \hat{\sigma}(x)dx} \epsilon_{bkg} + \int_0^D e^{-\int_0^x \hat{\sigma}(y)dy} \hat{\epsilon}(x)\, dx \tag{44}$$

Now identifying equivalent terms in 43 and 44 is straightforward. Both equations express the same, but use different notation.

$$\begin{aligned} \hat{L}(0) &\equiv I \\ \hat{\sigma} &\equiv \tau \\ \hat{\epsilon} &\equiv \epsilon \\ \epsilon_{\hat{bkg}} &\equiv I_b \\ x &\equiv t \\ y &\equiv u \end{aligned}$$

[6] It is just a change in notation of 42 to make more evident its equivalence to 43.

5 Computation of the optical model.

5.1 Computation Methods.

Remember the equation corresponding to the Single Scattering Model, equation 36:

$$I = \int_0^D g(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t + I_b \cdot \exp\left(-\int_0^D \tau(t)\delta t\right) \tag{45}$$

The explanations will be focused on how to compute the first addition term, being then straightforward the computation of the second one.

To compute the integral

$$\int_0^D g(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t\ , \tag{46}$$

it is first broken into pieces.

Let's be $\{t_1, \ldots, t_{n+1}\}$ a partition of the interval of integration, $[0, D]$, in n pieces $(n \geq 1)$. That is, $t_1 = 0$, $t_{n+1} = D$, and $t_1 < t_2 < t_3 < \ldots < t_{n+1}$.

$$\begin{aligned}
&\int_0^D g(t) \cdot \exp\left(-\int_0^t \tau(u)\delta u\right)\delta t = \\
&= \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} g(t) \cdot \exp\left(-\int_{t_1}^t \tau(u)\delta u\right)\delta t = \\
&= \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} g(t) \cdot \exp\left(-\int_{t_1}^{t_i} \tau(u)\delta u - \int_{t_i}^{t} \tau(u)\delta u\right)\delta t = \\
&= \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} g(t) \cdot \underbrace{\exp\left(-\int_{t_1}^{t_i} \tau(u)\delta u\right)}_{\text{respect to t, it is a constant}} \cdot \exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\delta t = \\
&= \sum_{i=1}^{n} (\underbrace{\int_{t_i}^{t_{i+1}} g(t) \cdot \exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\delta t}_{\text{(a)}} \cdot \underbrace{\exp\left(-\int_{t_1}^{t_i} \tau(t)\delta t\right)}_{\text{(b)}}) = \ldots
\end{aligned}$$

To shorten, let's call G_i to the term (a). The integral in (b) is again a sumation of integrals, and so (b) is a product of exponentials.

$$\ldots = \sum_{i=1}^{n} (G_i \cdot \underbrace{\prod_{j=1}^{i-1} \exp\left(-\int_{t_j}^{t_{j+1}} \tau(t)\delta t\right)}_{\text{(c)}}) = \ldots$$

And again to shorten, calling T_i (for transparency) to the term (c),

$$\ldots = \sum_{i=1}^{n}(G_i \prod_{j=1}^{i-1} T_j)$$

It is clear from the development made above that the second addition term in equation 45 also reduces to a product of exponentials. Thus, equation 45 has reduced to

$$I = (\sum_{i=1}^{n} G_i \cdot \prod_{j=1}^{i-1} T_j) + I_b \cdot \prod_{j=1}^{n} T_j \tag{47}$$

with

$$G_i = \int_{t_i}^{t_{i+1}} g(t) \exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\delta t \qquad \text{for } i = 1..n \tag{48}$$

$$T_j = \exp\left(-\int_{t_j}^{t_{j+1}} \tau(t)\delta t\right) \qquad \text{for } j = 1..n \tag{49}$$

The integrals in equations 48 and 49 may be computed numerically, or simplificative assumptions may be done that allow to compute them analytically. Whatever way they are computed, equation 47 expresses how the terms G_i and T_j must be composited; but it does not fix the order of the sumation. Equation 47 may be evaluated in Front to Back or in Back to Front orders. Front to Back requires an accumulator for transparencies (or alternatively, opacities), but it allows to stop the summation when the accumulated transparency is smaller (or alternatively, when the accumulated opacity is bigger) than some threshold value. Back to Front, however, does not need that accumulator, but the summation has to be made entirely.

A survey of numerical integration methods may be found in appendix C. Next, some of the simplifications usually made are explained. Of course, being simplifications, they introduce an error. The simplifications are labeled with letters from A to G.

A. Suppose that $g(t)$ is proportional to $\tau(t)$ in each segment (equation 26).

This allows the G_i to be computed analitically.
$g(t) = C_i\tau(t) \quad \forall t \in [t_i, t_{i+1}]$, with $C_i \in \mathbf{R}^+$ and for $i = 1..n$

$$\begin{aligned} G_i &= \int_{t_i}^{t_{i+1}} g(t) \cdot \exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\delta t = \\ &= \int_{t_i}^{t_{i+1}} C_i\tau(t) \cdot \exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\delta t = \\ &= C_i \int_{t_i}^{t_{i+1}} -\frac{\delta}{\delta t}\exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\delta t = \end{aligned}$$

$$\begin{aligned}
&= C_i \left[1 - \exp\left(-\int_{t_i}^{t} \tau(u)\delta u\right)\right]_{t_{i+1}}^{t_i} = \\
&= C_i \left(1 - \exp\left(-\int_{t_i}^{t_{i+1}} \tau(u)\delta u\right)\right) = \\
&= C_i\,(1 - T_i)
\end{aligned}$$

And then equation 47 may be rewritten as

$$I = \sum_{i=1}^{n} C_i(1 - T_i) \prod_{j=1}^{i-1} T_j \; + \; I_b \prod_{j=1}^{n} T_j \tag{50}$$

B. Only with the assumption A, the transparencies T may be computed in whatever way. Now, further assume that τ is constant in each segment.
$\tau(t) = \tau_i \qquad \forall t \in [t_i, t_{i+1}]$, with $\tau_i \in \mathbf{R}^+$ and for $i = 1..n$

$$\begin{aligned}
G_i &= \;\ldots.A\ldots. = C_i \left(1 - \exp\left(-\int_{t_i}^{t_{i+1}} \tau(u)\delta u\right)\right) = \\
&= C_i \left(1 - \exp\left(-\int_{t_i}^{t_{i+1}} \tau_i \delta u\right)\right) = \\
&= C_i(1 - \exp(-\tau_i(t_{i+1} - t_i))) \; = \; C_i(1 - \exp(-\tau_i l_i)) = \\
&= C_i(1 - T_i')
\end{aligned}$$

where l_i is defined as $l_i = t_{i+1} - t_i$ and is the length of the segment $[t_i, t_{i+1}]$ and T_i' is defined as $T_i' = \exp(-\tau_i l_i)$ and is the transparency of the segment $[t_i, t_{i+1}]$, under assumption B.

When an optical property is assumed to be constant along a ray segment, as it has been done for $\tau(t)$, what is usually done is to measure the property at some point $\tilde{t}_i$, $t_i \leq \tilde{t}_i \leq t_{i+1}$ of the segment. It is common to choose $\tilde{t}_i$ to be the middle point of the segment, or one of its extrems; but it may actually be whatever interior point. When the ray segments have all of them equal length, the points $\tilde{t}_i$ use to be equal-spaced.

Another possibility to set a constant value for an optical property inside a ray segment is to compute the property in the extrems of the segment, and average them, as pointed out in [WG91] and in [WM92]. For $\tau(t)$ in the interval $[t_i, t_{i+1}]$ this would be $\tau_i = \frac{1}{2}(\tau(t_i) + \tau(t_{i+1}))$. (What is the trapezoidal rule - see appendix C -).

Equation 47 rewrites as

$$I = \sum_{i=1}^{n} C_i(1 - T_i') \prod_{j=1}^{i-1} T_j' \; + \; I_b \prod_{j=1}^{n} T_j' \tag{51}$$

Defining $\alpha_i = 1 - T_i'$, the opacity of the i-th segment,

$$I = \sum_{i=1}^{n} C_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \; + \; I_b \prod_{j=1}^{n} (1 - \alpha_j) \tag{52}$$

This is the expression for the intensity along a ray used in [Lev88] and [Lev90]. This formulation is an itered application of the *over* operator of Porter and Duff [PD84].

For $n = 1$, $I \; = \; C_1\alpha_1 + I_b(1 - \alpha_1) \; = \; C_1\alpha_1 \; over \; I_b$ [7]

For $n = 2$, $\begin{aligned} I \; &= \; C_1\alpha_1 + C_2\alpha_2(1 - \alpha_1) + I_b(1 - \alpha_1)(1 - \alpha_2) \\ &= \; C_1\alpha_1 \; over \; C_2\alpha_2 \; over \; I_b \end{aligned}$

In general, for n, $I \; = \; C_1\alpha_1 \; over \; C_2\alpha_2 \; over \; C_3\alpha_3 \; over \; \ldots \; over \; C_n\alpha_n \; over \; I_b$

The *over* operator is not conmutative, but associative. In this way, equation 52 still accomodates both Back to Front and Front to Back evaluation orders.

$$\begin{aligned} I \; &= \; C_1\alpha_1 \; over \; C_2\alpha_2 \; over \ldots \; over \; C_n\alpha_n \; over \; I_b \; = \\ &= \; ((\ldots(C_1\alpha_1 \; over \; C_2\alpha_2) \; over \; \ldots) \; over \; C_n\alpha_n) \; over \; I_b \; = & \leftarrow\text{Front to Back} \\ &= \; C_1\alpha_1 \; over \; (C_2\alpha_2 \; over \; (\ldots \; over \; (C_n\alpha_n \; over \; I_b)\ldots)) & \leftarrow\text{Back to Front} \end{aligned}$$

C. Consider $g(t)$ is constant in each segment, not necessarily proportional to $\tau(t)$.

$g(t) = g_i \qquad \forall t \in [t_i, t_{i+1}]$, with $g_i \in \mathbf{R}^+$ and for $i = 1..n$

$$\begin{aligned} G_i \; &= \; \int_{t_i}^{t_{i+1}} g(t) \cdot \exp\left(- \int_{t_i}^{t} \tau(u)\delta u \right) \delta t = \\ &= \; g_i \int_{t_i}^{t_{i+1}} \exp\left(- \int_{t_i}^{t} \tau(u)\delta u \right) \delta t \end{aligned}$$

D. Furthermore considering C, suppose also that $\tau(t)$ is constant in each segment.

$\tau(t) = \tau_i \qquad \forall t \in [t_i, t_{i+1}]$, with $\tau_i \in \mathbf{R}^+$ and for $i = 1..n$

[7] William and Max in [WM92] describe this composition as $C_1\alpha_1$ *atop* I_b. The *atop* operator is another operator described in [PD84]. This statement is true only because the background is opaque, that is, $\alpha_b = 1$.

$$\begin{aligned}
G_i &= g_i \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_i}^{t} \tau_i \delta u\right) \delta t = \\
&= g_i \int_{t_i}^{t_{i+1}} \exp(-\tau_i \cdot (t - t_i)) \delta t = \\
&= \frac{g_i}{-\tau_i} \int_{t_i}^{t_{i+1}} -\tau_i \cdot \exp(-\tau_i \cdot (t - t_i)) \delta t = \\
&= \frac{g_i}{-\tau_i} [\exp(-\tau_i \cdot (t - t_i))]_{t_i}^{t_{i+1}} = \\
&= \frac{g_i}{-\tau_i} (\exp(-\tau_i \cdot l_i) - 1) = \\
&= \frac{g_i}{\tau_i} (1 - \exp(-\tau_i \cdot l_i)) = \\
&= \frac{g_i}{\tau_i} (1 - T_i')
\end{aligned}$$

Then equation 47 becomes

$$I = \sum_{i=1}^{n} \frac{g_i}{\tau_i} (1 - T_i') \prod_{j=1}^{i-1} T_j' + I_b \prod_{j=1}^{n} T_j' \tag{53}$$

This is done in [CCH95] and in [WG91]. Note that B is a particular case of D ($g_i = C_i \tau_i$).

As explained before, the values g_i, τ_i may be obtained sampling the ray segment. Even though usually both $g(t)$ and $\tau(t)$ are sampled in the same point of the segment, this is not strictly necessary.

If the sampled value of τ_i results to be $\tau_i = 0$, then $G_i = g_i \cdot l_i$.

E. The simplifications above consisted of setting $g(t)$, or $\tau(t)$, or both to constant values along each segment. Another possible simplificative assumption is to consider that these functions vary piecewise-linear along each segment ([WM92],[WG91]).

F. Functions $g(t)$ and $\tau(t)$ depend on the wavelength of light. Frequently they are considered to be three-valued functions

$$\tau(t) = \tau_\lambda(t) = (\tau_{red}(t), \tau_{green}(t), \tau_{blue}(t))$$

$$g(t) = g_\lambda(t) = (g_{red}(t), g_{green}(t), g_{blue}(t))$$

A simplificative assumption also possible consist of considering $\tau(t)$ to be independent of wavelength. In such a way $\tau(t)$ is a single-valued function, and less exponentials have to be computed ([WG91],[MHC90]).

To save computation effort is also possible, if assumption B has been made, to approximate the opacity $\alpha_i = 1 - \exp(\tau_i \cdot l_i)$ by $\min(1, \tau_i \cdot l_i)$ as in [WG91]. The

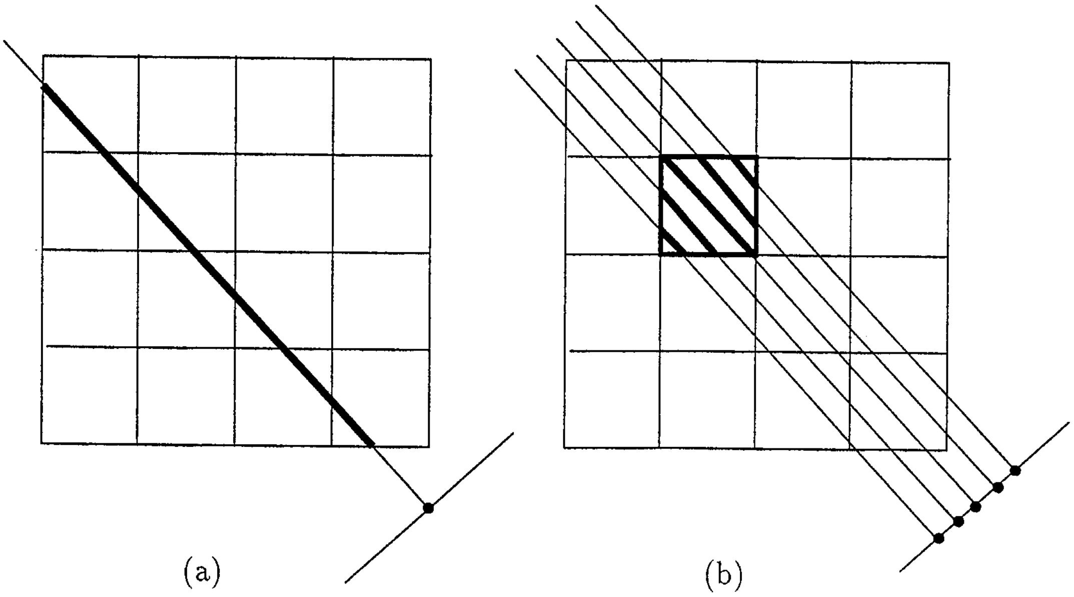


Figure 14: (a) ray casting methods; (b) projective methods.

mathematics behind this approximation is an application of Taylor's theorem, and may be found in appendix D.

G. Assumptions D and F are made in [WG91] and in [TPN95].

$$G_i = \frac{g_i}{\tau_i}(1 - T_i')$$

$$T_i' = \exp(-\tau_i l_i)$$

and $(1 - T_i')$ is approximated by $\min(1, \tau_i l_i)$.

The magnitude $\tau_i l_i$ is usually < 1; as a consequence, $G_i = \frac{g_i}{\tau_i} \cdot \tau_i l_i = g_i l_i$.

This is called by Wilhelms and Van Gelder, "color times distance" integration. Note that these assumptions are equivalent in the result to consider that there is no absorption inside the segment, thar is, $\tau_i = 0$ in the development of D. But that does not lead to an only emission model: what is assumed implicitly in these papers is that light emitted inside a segment (cell) is not attenuated in that segment but it is in the other segments or cells towards the observer.

Novins and Arvo, in [NA92], apply standart numerical integration methods for the computation of equation 46; bounds of the error are given for the particular case when functions $g(t)$ and $\tau(u)$ in the integrand are polynomials. If the integral has to be computed along a viewing ray with a prescribed accuracy δ, the integral at each voxel traversed by the ray is computed with accuracy $\frac{\delta}{n}$

(where n is the total number of cells the ray traverses). The authors present a more sophisticated evaluation method in [NAS92]. This method allows a major speed-up by setting the accuracy adaptatively along the ray. The philosophy is to concentrate the computations in the ray segments where the error may be largerly reduced.

5.2 Projection strategies.

To visualize a volume, two families of algorithms exist [8]:

- those that cast rays from the observer's eye through the volume (one or more rays per pixel), which are called ray casting methods.
- those that project cumulatively the cells of the volume on to the image plane, which are called projective methods.

As equations reported herein are physically based, as it has been seen, they are equally true whatever algorithm is used. Although looking the equation it seems that leads more naturally to ray casting algorithms, they also hold for projective methods and are also applied in them.

It was said in the previous section that, usually, to compute the integrals of the cumulative intensity in the emission-absorption model, these integrals were broken down into pieces. In this way, the path $[0, D]$ through the volume was partitioned in n pieces:

$$[t_1, t_2], [t_2, t_3], \cdots, [t_{n-1}, t_n], [t_n, t_{n+1}]$$

being $t_1 = 0 < t_2 < t_3 < \ldots < t_{n-1} < t_n < t_{n+1} = D$, and being l_i defined as the length of the i-th segment, that is, $l_i = t_{i+1} - t_i$.

Usually, two kinds of partition may be made. The first one is an equidistant partition, that is, all the n segments have the same length; $t_i = (i-1) \cdot \frac{D}{n}$ and $l_i = \frac{D}{n}$. The second one breaks the ray into the segments intersected by the cells of the volume, so the partition is not equidistant.

In ray casting algorithms, it is possible to partition the ray equidistantly ([Lev88],[Lev90],[Sab88]). or to partition it accordingly to cells intersections ([UK88]). In projective algorithms, however, as the process is made cell by cell, integral may be partitioned only at the extent of the cells that form the volume representation ([MHC90], [WG91], [WM92]). In these algorithms, the values of each projection are related with the light transfer integrals along view directions through the pixels of the projection and inside the cell extent, while the cell compositing is related to the combination of the integral pieces to give the final intensity.

[8] These two families of algorithms also receive the names of "image order" and "object order", respectively.

In this way, ray casting and projective methods only differ in the order the computations are made. In ray casting the lighting model is computed entirely along a ray, one ray at a time. In projective methods several integrals are computed simultaneously and incrementally, as the cells are being projected, and the intermediate results are stored in an image buffer. See figure 14. At least theoretically, both kinds of methods should report the same image results.

Next, an explanation of the projective methods in papers [MHC90], [WG91] and [WM92] follows. In [MHC90], Max et al. suppose an arrangement of convex volume polyhedra (i.e. tetrahedra), eventually separated by contour polygons. Polyhedra and polygons are sorted in depth, and projected on the screen, either back to front or front to back, compositing the color and the opacity of the individual elements as in 47. To compute such color and opacity of each volume polyhedron, the emission-absorption model is used, assuming that g is related to τ by a constant inside each polyhedron and assuming also that τ is an analitically integrable function. This allows the integrals in 48 and 49 to be calculated not only analitically, but also incrementally inside each convex polyhedron.

Wilhelms and van Gelder also exploits coherence in cell projection in a similar fashion [WG91]. Noticing that the orthogonal projections of the cells of a regular grid have all the same shape, the scan conversion of the faces of a cell need to be made only once. A model cell is subdivided into subcells, pieces that have the same front and back faces and which projections are not crossed by any projected edge. Depending on the point of view, a cell may be subdivided into one up to seven subcells. Each subcell projects onto a polygon in the image space. For each cell, color and opacity are computed by integration in depth for the vertexs of those polygons. Several simplificative assumptions may be done in that computation. Finally the computed values are bilinearly interpolated across the projection, and composited either front to back or back to front. Interpolating optical properties instead of the scalar field, Wilhelms and van Gelder lose in image quality; in Max's words [Max95]: "Interpolating f first permits the optical properties to change rapidly within a single volume element, to emphasize a small range of scalar values. It is possible to compute the optical properties only at the grid vertices and then interpolate them instead, but this may eliminate fine detail. This situation is analogous to the superiority of Phong shading (interpolating the normal) over Gouraud shading (interpolating the shading color) for representating fine highlight detail." The authors sacrifice some of the image quality by the speed-up achieved by using specialized hardware devoted to Gouraud shading of polygons.

Similar ideas of those exposed in [WG91] are presented in [WM92] by Williams and Max. Emission and absorption model is employed in this paper to derive an expression for the integration in depth of color and opacity inside a volume cell. Some assumptions made in [WM92] coincide with those in [WG91], such as approximating the opacity $\alpha = 1 - \exp -\tau_i l_i$ by $\tau_i l_i$. But Williams and Max differ from Wilhems and van Gelder in that the former consider the source term

$g(t)$ depend linearly on the optical density $\tau(t)$, $(g(t) = C(t)\tau(t))$. The resultant integral of their model has closed form solution when the transfer functions that map the scalar field to the optical properties $\tau(t)$ and $C(t)$ are linear or piecewise linear, and the scalar field within the cell is integrable.

It is understood that in all these three methods, the more pixels a cell projects on, the more advantage is reported by coherence.

5.3 Surface visualization.

In volume rendering, if contribution of all cells is considered using a volume scattering term, or emission and absorption only, the volume appears in the image as a kind of cloud, nebulous jelly. Such a projection of all the information contained into the volume data may result difficult to be interpreted. It is sometimes preferable, as it has already been said, to visualize just a part of the volume data: isovalued surfaces.

An isovalued surface, or isosurface, is the set of all points in the space that share the same value of the scalar field. To achieve the visualization of an arbitrary surface, two main approaches exist: the explicit extraction of geometric primitives that approximate the isosurface and the visualization of the isosurface 'on the fly', directly from volume representation. The former approach is called *isosurface reconstruction* and requires two steps: surface extraction, either with the *cuberille* method [CHRU85], the *marching cubes* [LC87], ..., and surface shading. The shading is done with any of the standart methods exposed in section 3. Moreover, specific shading methods have been developed for cuberille surfaces, both in object and image spaces, as reported in [CHRU85]. In the latter approach, which receives the name of *direct volume rendering*, also two steps are performed: surface detection and surface shading. Surface shading is a common step in both approaches, and it implies the estimation of the surface normal.

In the next sections, the steps for direct volume rendering will be tackled.

5.3.1 Surface detection in direct volume rendering.

Similar results to the ones of reconstructed surface visualization may be obtained without the explicit construction of any geometric mesh, simply by the application of the single scattering equation with a transfer function for τ that maps scalar values to $\tau = 0$ to $\tau = +\infty$ according to a binary classification. Figure 15 shows examples of such a mapping.

It should be noted that due to the caption device limited precision and due to noise, it is difficult to find an arbitrary classification function to accurately represent and object surface. It possible to miss voxels that belong to a surface and/or accept others that do not. This translates, when visualizing, to loss of detail, holes on surfaces, and spureous voxels.

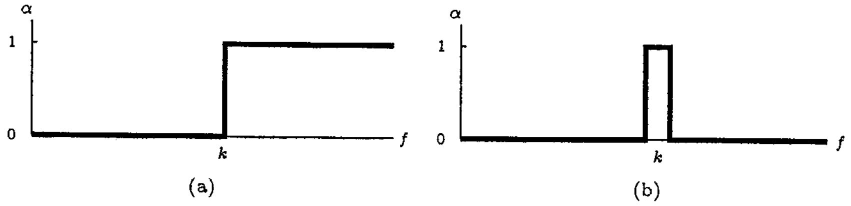


Figure 15: Mappings from f to τ have been substituted by mappings from f to α, because of the difficulty of representing $+\infty$. (a) threshold mapping (b) window mapping.

A solution is to consider probabilistic answers to the classification process, instead of binary ones. Three approaches will be cited here that rely on probabilistic classification methods and that deal with biomedical CT data. The CT measurement of human tissues (bone, sofgt tissue and fat)[9] does not result on a single characteristic value for each tissue, but a range of values around a characteristic one. The range of tissues overlap, and so for a given measured value is not always clear what tissue it corresponds to. To visualize the surface between tissues, it is of great importance to be able to classify correctly each sampled point.

[Lev88] and [DCH88] make a simplificative assumption about possible tissue adjacencies in the volume data: at each location any tissue type may touch at most two other types and, if tissue types are ordered by CT number, the only possible adjacencies are between consecutive types in the ordering.

Be f_1, f_2, f_3 and f_4 four material characteristic values, and be α_1, α_2, α_3 and α_4 the opacities assigned to these materials respectively. Any value of f either coincide with one of the characterisitic values or is placed between two of them. When a value $f(x)$ is between the characteristic values f_i and f_{i+1}, Levoy, in [Lev88], assigns to it an opacity by interpolating between α_i and α_{i+1}.

$$\alpha(x) = \begin{cases} |\nabla f(x)| \cdot \left(\alpha_{i+1} \cdot \frac{f(x)-f_i}{f_{i+1}-f_i} + \alpha_i \cdot \frac{f_{i+1}-f(x)}{f_{i+1}-f_i}\right) & \text{if } f_i \leq f(x) \leq f_{i+1} \\ & \text{and for } i = 1..3 \\ 0 & \text{otherwise} \end{cases} \tag{54}$$

The gradient of f at point x is used as a surface presence detector; it is also used for this purpose in [DCH88] and in [CCH95]. Multiplying by the gradient, homogeneous regions are set to transparent or almost transparent, while surfaces are emphasized. The quantities $\frac{f(x)-f_i}{f_{i+1}-f_i}$ and $\frac{f_{i+1}-f(x)}{f_{i+1}-f_i}$ may be understood as

[9] Air is also sampled in the data, corresponding either to the exterior of the body, either to the interior of the lungs.

probabilities of value $f(x)$ to belong to materials $i+1$ and i, respectively. With Levoy's method what is displayed is not a thin isosurface but rather a thick boundary region that allows fine detail to be displayed, although with small opacity.

The segmentation of Levoy is based only onto the single characteristic values f_i. A more accurated predictor may be made by using the value distribution functions of each material, that are known a priori. Drebin et al. in [DCH88] use such distributions to better estimate the percentage of a material inside a cell. They use a Bayesian estimator.

$$P(i|f(x)) = \frac{P(f(x)|i)}{\sum_{j=1}^{4} P(f(x)|j)} \tag{55}$$

where $P(i|f(x))$ is the probability that a value $f(x)$ were obtained from material i, and $P(f(x)|j)$ is the probability that material j had a value $f(x)$ which may be known a priori. $P(i|f(x))$ estimates the percentage of a voxel with value $f(x)$ occupied by material i. Once $P(i|f(x))$ is determined for all materials, the optical properties of the voxel are computed by averaging characteristic colors and opacities of the materials, according to their presence percentages.

Both [Lev88] and [DCH88] only allow mixtures of consecutive materials in the CT ordering. They can not determine mixtures of separate materials as fat and bone for instance, that actually may appear in data. Cai et al. [CCH95] propose a new and more sophisticated probabilistic segmentation that deals with these cases correctly and allows up to three materials adjacencies in one voxel. Unlike [Lev88] and [DCH88], furthermore considering the value of f and its distribution, the gradient of f is also taken into account in the segmentation process. The probability that the current sample belongs to one material is a two dimensionmal Gauss distribution function of f and its gradient.

$$P(X|i) = \frac{1}{2\pi\sqrt{|C|}} \cdot e^{-\frac{1}{2}(X-M_i)^T \cdot C^{-1} \cdot (X-M_i)} \tag{56}$$

where X is a vector $(f(x), \nabla f(x))$, M_i is the peak position of the gaussian distribution of X, and C is the covariance matrix between f and ∇f. For each sampled value, the corresponding probabilities are calculated for all material. The probabilities that exceed some threshold indicate the presence of such material in the voxel. A surface exists when more than one material is present in the voxel. To project the volume they use a ray casting scheme, and shading is done by what they call the Composed Scattering Model: a single scattering term that is a weighted sumation of surface scattering and volume scattering. The corresponding weights are computed from the probabilities supplied by the segmentation scheme. The combination of this sophisticated segmentation method and the composed scattering produce high quality images. The Composed Segmentation Model is subsumed by the optical model of equation 36, by expanding the term $g(t)$ as

$$g(t) = \epsilon(t) + k_1 S_s(t) + k_2 S_v(t) \tag{57}$$

5.3.2 Surface shading in direct volume rendering.

Once a point on a surface has been detected, its f value estimated, and it has been classified, it is shaded using a surface scattering term as explained in section 4.4.3. This computation requires the surface normal at the point, which is approximated by the gradient vector at that point.

In cubical or hexahedral grids, the gradient at a corner of each cell is calculated either as backward, forward, or central differences [10]. For each point inside a cell, the gradient at that point is calculated by trilinear interpolation of the gradient vectors of the cell's corners.

In linear tetrahedral grids, as the scalar field varies linearly inside each tetrahedron, the isosurfaces are planar and the gradient vector is constant inside each cell. To compute the gradient vector at each vertex of the grid is done by averaging the gradients of the cells concurrent at that vertex.

6 Conclusions.

This report addresses lighting models and shading algorithms for Volume Rendering.

Light theory has been briefly reviewed, showing that the particle model is the more suitable for Computer Graphics.

Light transport mechanisms through volumes have been described. In the bibliography, different derivations of the light transport equation exist using different, though equivalent, notations. An attempt has been made herein to unify these notations and show the equivalence between the optical model of [Max95] and the general light transport theory exposed in [Arv93].

It has also been found that for most computer graphics purposes. the general light transport model is considered too complicated and simplified by limiting

[10] In [DCH88] gradient is computed as forward differences, but what is most common is to calculate it as central differences, [Lev88].

Vector ∇f, gradient of f, at point (x_i, y_j, z_k) is a three component vector

$$\nabla f_x = \frac{f(x_{i+1}, y_j, z_k) - f(x_{i-1}, y_j, z_k)}{2S_x}$$
$$\nabla f_y = \frac{f(x_i, y_{j+1}, z_k) - f(x_i, y_{j-1}, z_k)}{2S_y}$$
$$\nabla f_z = \frac{f(x_i, y_j, z_{k+1}) - f(x_i, y_j, z_{k-1})}{2S_z},$$

where S_x, S_y and S_z are the distances between neighboring samples in the x, y and z directions, respectively.

to one the number of scattering events per particle. This simplified model is called the Single Scattering Model.

The analysis of the existing rendering litterature reveals the existence of a gap between the light transport model and practical shading algorithms. In order to fill this gap, the underlying assumptions of the most relevant shading algorithms have been exposed, both in ray tracing and projective approaches.

Many open problems exist in volume shading. First, as full volume visualization may be confusing, the outlying of relevant features such as isosurfaces is of great importance for a correct understanding of the data. Features characterization requires an accurate and effective segmentation process. In addition, a classification or labelling of the features is essential in order to compute optical properties allowing to simulate the desired behaviour of the features with light. Research on this topic is needed, as the existing methodologies still mostly rely on manual processes.

The computation of the single scattering equation is the kernel of the shading process. Its efficiency is crucial for the interactivity of the rendering. The study of acceleration techniques for this process without loss of precision is therefore essencial.

Finally, although the Single Scattering model may be applied both to ray casting and projective methods, the error metric in the latter case has not been fully analyzed. A promising research line is the investigation of a splatting strategy that could project voxels of different size and should be equivalent in the result to a ray casting method with any prescribed accuracy.

A Glossary of terms used in surface shading models (tables 1 and 2).

I light intensity, or color

I_a ambient light intensity

n number of point light sources

I_i light intensity of the i-th light source

S_i normalized vector along the direction from the point where shading is computed to the i-th light source

N surface normal at the point where shading is computed

k_a ambient-reflection coefficient

k_d diffuse-reflection coefficient

k_s specular-reflection coefficient

k_e specular-reflection exponent

R_i perfectly specular reflection direction of the i-th light source

V viewing direction (from the shaded point to the viewer)

H_i half way vector corresponding to the i-th light source (in the middle between S_i and V)

f_{att_i} light source attenuation of the i-th light source

$$f_{att_i} = min\left(\frac{1}{c_1 + c_2 d_i + c_3 {d_i}^2}\ ,\ 1\right)$$

where d_i is the distance from the shaded point to the i-th light source, and c_1, c_2, c_3 are user-defined constants

R_a ambient-reflection coefficient

s specular-reflection coefficient

d diffuse-reflection coefficient
$(s + d \leq 1)$

f_d diffuse term ($f_d = \frac{1}{\pi}$)

f_{s_i} specular term for the i-th light source

$$f_{s_i} = \frac{1}{\pi}\,\frac{F \cdot D \cdot G_i}{(N \cdot S_i)(N \cdot V)}$$

where,

D, distribution term, describes the microfacet's slope distribution function, (that is. the shape of the grooves). It is a kind of roughness measure. Different expressions for G have been given by several authors

G_i, geometry term, or geometric attenuation, measures the effects of shadowing and masking, that is, the block of light towards microfacets and from microfacets to the viewer produced by the surface geometry itself.

$$G_i = min\left(1\ ,\ \frac{2(N \cdot H_i)(N \cdot V)}{V \cdot H_i}\ ,\ \frac{2(N \cdot H_i)(N \cdot S_i)}{V \cdot H_i}\right)$$

F, Fresnel term, computes the amount of reflectance as a function of the wavelength of incident light and its angle of incidence

$\delta\omega$ differential solid angle, with extrem at the i-th light source and covering the surface area where the shading is computed

g_i geometric term to include shadows caused by the i-th light source (it is 0 if some object occludes the light in its path from the source to the shaded surface point, 1 otherwise). This term may appear explicitly, or may be embeded into the correspondent attenuation function.

I_s light intensity coming from specular reflection direction

k_t specular-transmission coefficient

I_t light intensity coming from specular transmission direction

B_i radiosity of patch i

B_j radiosity of patch j

E_i the rate at which light is emmitted from patch i

ρ_i reflectivity of patch i

$F_{i,j}$ form factor of patches i and j; it specifies the fraction of energy leaving patch i that arrives at patch j

N_p number of patches the scene is discretized on

B Ordinary Differential Equations.

Herein a very brief remainder of ODEs is presented, in order to facilitate the understanding of explanations in section 4.4.

Definition : An ODE of *n order* is an equation in the n-th derivative.

$$F(x,y,y',\ldots,y^{(n)})=0$$

$$F(x,y,y')=0 \qquad \text{first order ODE}$$

Definition : The *grade* of an n order ODE is the exponent (K) of the n-th derivative. If $K=1$ it is a *linear* ODE.

B.1 Linear First Order ODEs.

- $M(x,y)\,dx \;+\; N(x,y)\,dy \;=\; 0$
  is called *exact* ODE if $\frac{\delta M}{\delta y}=\frac{\delta N}{\delta x}$.
  In this case, is possible to find a solution integrating term by term.
- If $\frac{\delta M}{\delta y}\neq\frac{\delta N}{\delta x}$ the ODE is *not exact.*
- When the ODE is not exact, it is possible to try to find an *integrating factor.* An integrating factor is a function μ which satisfies that multiplying the ODE by the integrating factor it becomes exact.
- Some cases that makes it possible to find an integrating factor:

  1. $$\frac{\frac{\delta M}{\delta y}-\frac{\delta N}{\delta x}}{N}=f(x)$$

     ($f(x)$ function that only depends on x). In this case, the integrating factor is $\mu=e^{\int f(x)dx}$.

  2. $$\frac{\frac{\delta M}{\delta y}-\frac{\delta N}{\delta x}}{M}=g(y)$$

     ($g(y)$ function that only depends on y). In this case, the integrating factor is $\mu=e^{\int -g(y)dy}$.

C Numerical Integration Methods.

Definition. A *quadrature formula* is an expression such as

$$I_{n+1}\{f\} = \sum_{k=0}^{n} \alpha_k f(x_k)$$

that approximates numerically the integral $\int_a^b f(x)\delta x$.

The points x_k are called *nodes*, and are choosed in $[a,b]$, equidistantly or not. The values α_k are called *coefficients* or *weights*. Once determined the nodes, the weights, and the values of f at the nodes, the quadrature formula is completely determined.

Definition. The *precision* of a quadrature formula $I_{n+1}\{f\} = \sum_{k=0}^{n} \alpha_k f(x_k)$ in $[a,b]$ is the maximum m such that

$$\forall i = 0..m \;\; I_{n+1}\{x^i\} = \int_a^b x^i \delta x$$

That is, a quadrature formula has precision m if it integrates exactly all the polynomials of degree $\leq m$.

In this appendix, some quadrature formulas will be examined.

C.1 Riemann sum.

Divide the interval $[a,b]$ into $n{+}1$ subintervals, using points $\xi_1, \xi_2, \ldots \xi_n$, in (a,b) arbitrarily choosed. In each interval $[a,\xi_1]$, $[\xi_1,\xi_2]$, ... , $[\xi_n,b]$ choose points x_0, x_1, ... , x_n also arbitrarily. Defining, $\xi_0 = a$, $\xi_{n+1} = b$ and $\Delta\xi_k = \xi_{k+1} - \xi_k$, the expression

$$\sum_{k=0}^{n} \Delta\xi_k f(x_k)$$

is called a *Riemann sum* (geometrically, it is the total area of the rectangles in figure 16).

Increase n in such a way that each $\Delta\xi_k \to 0$. If the limit of the Riemann sums exists and does not depend on the way the x_k have been choosed, then this limit is $\int_a^b f(x)\delta x$. In this case is said that f is integrable or Riemann integrable [Spi69].

A simple numerical method to "calculate" an integral consists of to approximate it by a Riemann sum.

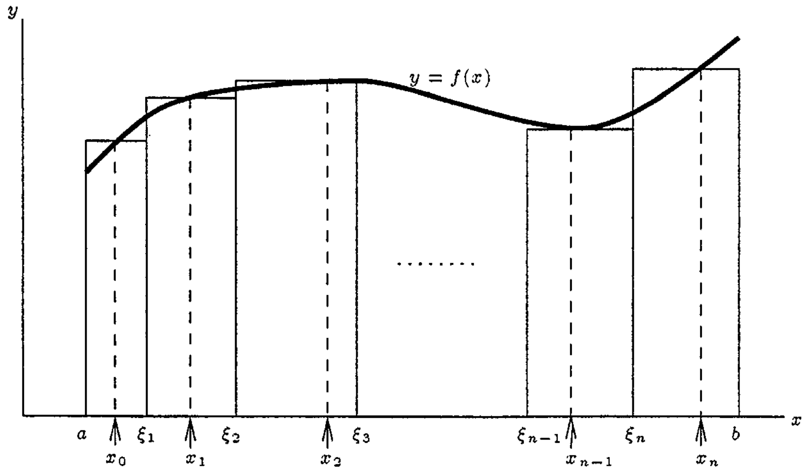


Figure 16: An n terms Riemann sum for function $f(x)$ in the interval $[a, b]$.

C.2 Newton-Cotes formulas.

This kind of quadrature formulas consists of to substitute the integrand f by an interpolant polynomial in an equidistant partition.

Define

$$P_n(x) := \sum_{k=0}^{n} y_k \cdot L_k(x) ,$$

where

$$y_k := f(x_k)$$

$$L_k(x) := \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}$$

(Lagrange interpolation), and the x_k points form an equidistant partition of $[a, b]$, with step $h = \frac{b-a}{n}$.

With the following change of variable, $x = a + th$ and $x_i = a + ih$, results

$$L_k = \varphi_k(t) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{t - i}{k - i}$$

n	C_k^n	Error[11]	Name
1	$\frac{1}{2}\frac{1}{2}$	$h^3\frac{1}{12}f^{\mathrm{II}}(\xi)$	Trapezoidal rule
2	$\frac{1}{6}\ \frac{4}{6}\ \frac{1}{6}$	$h^5\frac{1}{90}f^{(\mathrm{IV})}(\xi)$	Simpson rule
3	$\frac{1}{8}\ \frac{3}{8}\ \frac{3}{8}\ \frac{1}{8}$	$h^5\frac{3}{80}f^{(\mathrm{IV})}(\xi)$	3/8 rule
4	$\frac{14}{90}\ \frac{64}{90}\ \frac{24}{90}\ \frac{64}{90}\ \frac{14}{90}$	$h^7\frac{8}{945}f^{(\mathrm{VI})}(\xi)$	Milne's rule
5	$\frac{95}{288}\ \frac{375}{288}\ \frac{250}{288}\ \frac{250}{288}\ \frac{375}{288}\ \frac{95}{288}$	$h^7\frac{275}{12096}f^{(\mathrm{VI})}(\xi)$	-
6	$\frac{41}{840}\ \frac{216}{840}\ \frac{27}{840}\ \frac{272}{840}\ \frac{27}{840}\ \frac{216}{840}\ \frac{41}{840}$	$h^9\frac{9}{1400}f^{(\mathrm{VIII})}(\xi)$	Weddle's rule

Table 3: Some common Newton-Cotes formulas.

and

$$P_n = \beta_n(t) = \sum_{k=0}^{n} y_k \cdot \varphi_k(t)$$

As it has been said, the integration method consists of substituting f by an interpolator.

$$\int_a^b f(x)\delta x \approx \int_a^b P_n(x)\delta x = \int_a^b \sum_{k=0}^{n} y_k L_k(x)\delta x = \int_0^n \sum_{k=0}^{n} y_k \varphi_k(t)\delta x =$$

$$= \int_0^n \sum_{k=0}^{n} y_k\varphi_k(t)h\delta t = h\sum_{k=0}^{n} y_k \int_0^n \varphi_k(t)\delta t = nh\sum_{k=0}^{n} y_k \frac{1}{n}\int_0^n \varphi_k(t)\delta t =$$

$$= (b-a)\sum_{k=0}^{n} y_k \frac{1}{n}\int_0^n \varphi_k(t)\delta t = (b-a)\sum_{i=0}^{n} y_i C_k^n$$

The expressions $C_k^n = \frac{1}{n}\int_0^n \varphi_k(t)\delta t$ $(0 \leq k \leq n)$ does not depend on f, a nor b, and they may be precalculated for different values of n, as it is shown in table 3.

Define $E_{n+1}\{f\}$ as

$$E_{n+1}\{f\} := I_{n+1}\{f\} - \int_a^b f(x)\delta x$$

That is, $E_{n+1}\{f\}$ is the error committed by the quadrature formula $I_{n+1}\{f\}$, which is a quadrature formula of $n+1$ nodes.

In general, the expression of the error may be derived using the following two theorems:

[11] The indicated derivative of f must exist and be continuous. ξ some point in (a, b).

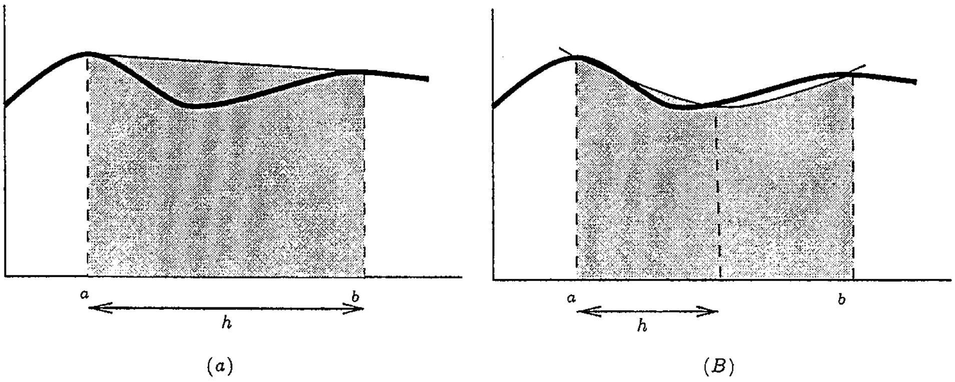


Figure 17: (a) Trapezoidal rule. The function is substituted by a straight line. $I_2 = \frac{b-a}{2}(f(a)+f(b))$. (b) Simpson rule. The function is substituted by a parabole. $I_3 = \frac{b-a}{6}(f(a)+4f(a+h)+f(b))$. The shadded area is the approximation calculated by the quadrature formula.

Theorem. If n is odd and f is C^{n+1} in $[a,b]$, then

$$E_{n+1}\{f\} = \frac{M_n}{(n+1)!}h^{n+2}f^{(n+1)}(\xi) \qquad \text{for some } \xi \text{ in } (a,b)$$

where $M_n := -\int_0^n t(t-1)\cdots(t-n)\delta t \ > 0$

Theorem. If n is even and f is C^{n+2} in $[a,b]$, then

$$E_{n+1}\{f\} = \frac{M_n}{(n+1)!}h^{n+3}f^{(n+2)}(\xi) \qquad \text{for some } \xi \text{ in } (a,b)$$

where $M_n := -\int_0^n t^2(t-1)\cdots(t-n)\delta t \ > 0$

(The reader may check the column of the error in table 3.)

If n is even, the corresponding Newton-Cotes formulas with $n+1$ and $n+2$ nodes have the same precision: n+1. So, when possible, is preferable to choose the first formula, because with one node less achieves the same precision[12]. Simpson rule is preferable to 3/8 rule, and Milne's rule to the following formula in table 3.

In figure 17 the well known trapezoidal and Simpson rules are illustrated. Other integration rules of this kind may be found by using another type of interpolation, instead of Lagrange interpolation, i.e. Hermite interpolation.

[12] This does not mean that the two formulas have the same error, but the difference is not significative.

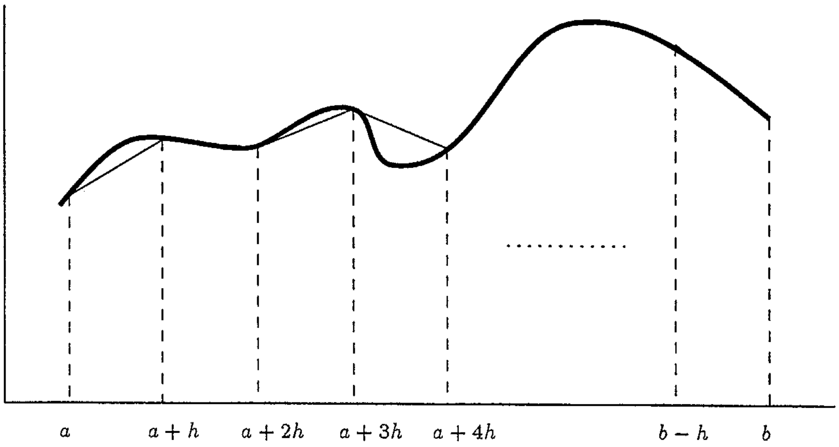


Figure 18: Composited trapezoidal rule.

C.3 Composited Newton-Cotes formulas.

When the interval of integration $[a, b]$ is large, the Newton Cotes formulas that have been seen do not make a good approximation, as the error depends directly on $h = \frac{b-a}{n}$.

Trying to increase n, that is to choose a Newton-Cotes formula that uses a higher degree polynomial interpolator, neither works, because high degree interpolators, although interpolate f at nodes, vary wildly between them.

What may be done to integrate a function correctly in a large interval, is to subdivide it into several subintervals, to apply a Newton-Cotes formula in each interval, and to add the results. This gives rise to the composited Newton-Cotes formulas.

Here, only composited trapezoidal and Simpson rules will be explained. The generalization to other composited formulas is straightforward.

C.3.1 Composited trapezoidal rule.

Divide the integration interval into N subintervals of equal length. Apply the trapezoidal rule to each subinterval, adding the results. It is shown in figure 18; $h = \frac{b-a}{N}$, $N \in \mathrm{N}$.

Calling $T(h)$ to the total sum, and $T_i(h)$ to the trapezoidal rule in the interval $[a + ih, a + (i + 1)h]$, results

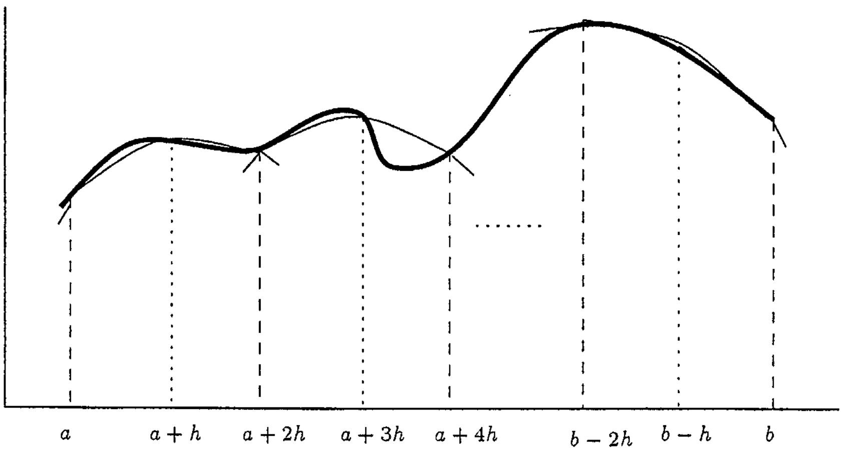


Figure 19: Composited Simpson rule.

$$T(h) = \sum_{i=0}^{N-1} T_i(h)$$

and

$$T_i(h) = \frac{h}{2}\left(f(a+ih) + f(a+(i+1)h)\right)$$

Then

$$\begin{aligned} T(h) &= \frac{h}{2}\left(f(a) + f(a+h)\right) + \frac{h}{2}\left(f(a+h) + f(a+2h)\right) + \ldots \\ &\quad \ldots + \frac{h}{2}\left(f(a+(N-1)h) + f(b)\right) \end{aligned}$$

$$T(h) = \frac{h}{2}\left[\, f(a) + 2f(a+h) + 2f(a+2h) + \ldots + 2f(b-h) + f(b)\right]$$

If the function f is C^2 in $[a,b]$, then the error committed when applying the composited trapezoidal rule is

$$T(h) - \int_a^b f(x)\delta x = \frac{b-a}{12}h^2 f''(\xi) \qquad \text{for some } \xi \text{ in } (a,b).$$

C.3.2 Composited Simpson rule.

Divide the integration interval into N subintervals of equal length. Apply the Simpson rule to each subinterval, adding the results. It is shown in figure 19; $h = \frac{b-a}{2N}$, $N \in \mathrm{N}$.

Calling $S(h)$ to the total sum, and $S_i(h)$ to the Simpson rule in the interval $[a + ih, a + (i + 1)h]$, results

$$S(h) = \sum_{i=0}^{N-1} S_i(h)$$

and

$$S_i(h) = \frac{h}{3}\left(f(a + 2ih) + 4f(a + (2i + 1)h) + f(a + (2i + 2)h)\right)$$

Then

$$\begin{aligned} S(h) &= \frac{h}{3}\left(f(a) + 4f(a + h) + f(a + 2h)\right) + \\ &\quad \frac{h}{3}\left(f(a + 2h) + 4f(a + 3h) + f(a + 4h)\right) + \ldots \\ &\quad \ldots + \frac{h}{3}\left(f(a + (2N - 2)h) + 4f(a + (2N - 1)h) + f(b)\right) \end{aligned}$$

$$\begin{aligned} S(h) &= \tfrac{h}{3}\ \left[\ f(a) + 4f(a + h) + 2f(a + 2h) + 4f(a + 3h) + \ldots \right. \\ &\quad \left. +2f(b - 2h) + 4f(b - h) + f(b)\ \right] \end{aligned}$$

If the function f is C^4 in $[a, b]$, then the error committed when applying the composited Simpson rule is

$$S(h) - \int_a^b f(x)\delta x = \frac{b-a}{180}h^4 f^{(\mathrm{IV})}(\xi) \qquad \text{for some } \xi \text{ in } (a.b).$$

C.4 Euler-McLaurin summation formula.

If f is C^{2m+2} in $[a, b]$, and $T(h)$ is the composited trapezoidal rule of N sumands in $[a, b]$, the Euler-McLaurin summation formula states that:

$$\begin{aligned} T(h) = \int_a^b f(x)\delta x \quad &+ \quad \sum_{l=1}^{m} h^{2l} \frac{B_{2l}}{(2l!)} \left[f^{(2l-1)}(b) - f^{(2l-1)}(a) \right] \\ &+ \quad h^{2m+2} \frac{B_{2m+2}}{(2m + 2)!}(b - a) f^{(2m+2)}(\xi) \end{aligned}$$

where $h = \frac{b-a}{N}$
and the B_k are the Bernouilli's numbers, that may be defined by the recursive formula

$$B_0 = 1$$
$$B_n = \sum_{k=0}^{n} \binom{n}{k} B_k$$

The expression for $T(h)$ is an assimptotic expansion of the composited trapezoidal rule in terms of even powers of h.

$$T(h) = C_0 + C_1h^2 + C_2h^4 + C_3h^6 + \ldots + \bigcirc(h^{2m+2})$$

$T(h)$ is a first approximation to the unknown integral, C_0. If a better approximation is desired, some terms of the expansion may be added to $T(h)$.

C.5 Romberg's method.

The expression $T(h) = C_0 + C_1h^2 + \ldots + \bigcirc(h^{2m+2})$ states that the error committed in a composited trapezoidal rule mainly depends on h.

Cutting in the first term,

$$T(h) = C_0 + \bigcirc(h^2)$$

This means that when dividing by two the step h, it is doubling the number of trapezoids, the error reduces in the order of four times.

$$T(h) = C_0 + \epsilon$$

$$T\left(\frac{h}{2}\right) \approx C_0 + \frac{\epsilon}{4}$$

It follows then that

$$T\left(\frac{h}{2}\right) - T(h) \approx -\frac{3}{4}\epsilon$$

$$\frac{T(\frac{h}{2}) - T(h)}{3} \approx \frac{-\epsilon}{4}$$

As $C_0 \approx T(\frac{h}{2}) - \frac{\epsilon}{4}$, results that the expression

$$T\left(\frac{h}{2}\right) + \frac{T(\frac{h}{2}) - T(h)}{3}$$

aproximates the integral $C_0 = \int_a^b f(x)\delta x$ closer than $T(h)$ and $T(\frac{h}{2})$.

This is known as Richardson extrapolation.

The Romberg's method consists of to apply Richardson extrapolation iteratively.

Romberg's method.
Be h_0 an initial step $\left(\frac{b-a}{h_0} \in \mathrm{N}\right)$, and define $h_i := \frac{h_{i-1}}{2}$ for $i \geq 1$.

Define $T_{i\,0}$ as $T_{i\,0} := T(h_i)$, the trapezoidal sum with step h_i.

The successive refinements are found by the recursion

$$T_{i\,k} = T_{i\,k-1} + \frac{T_{i\,k-1} - T_{i-1\,k-1}}{4^k - 1}$$

and may be placed in a triangular matrix.

	$k=0$	$k=1$	$k=2$	$\ldots$
$i=0$	$T(h_0) = T_{0\,0}$			
$i=1$	$T(h_1) = T_{1\,0}$	$T_{1\,1}$		
$i=2$	$T(h_2) = T_{2\,0}$	$T_{2\,1}$	$T_{2\,2}$	
$\vdots$		$\vdots$		

The process goes on adding files to the matrix and it stops when two consecutive values $T_{i\,k}$ and $T_{i+1\,k}$ are close enough each other. Either $T_{i+1\,k}$ or $T_{i+1\,k+1}$ may then be choosed as a final approximation, as desired (both values will be very similar).

In practice, more than seven colums are never done; if in the seventh column the desired accuracy has not been achieved, the process continues adding files with no more than seven components.

C.6 Gaussian quadrature formulas.

Up to now, equidistant nodes have been considered. It has been seen that with n equidistant nodes, the maximum precision that may be achieved is at most n. If the nodes are allowed to be not equally spaced, then the maximum precision reachable with n nodes is $2n-1$, and it is achieved using Gaussian quadrature formulas.

The Gaussian quadrature formulas solve the integrals of the kind

$$\int_a^b \omega(x) f(x) \delta x$$

where $\omega(x) \geq 0$ is a continuous function in (a, b), and the interval $[a, b]$ may be infinite ($[0, +\infty]$ or $(-\infty, +\infty)$).

Definition. Define the scalar product $< f, g >$ respect to $\omega(x)$ in $[a, b]$ as

$$< f, g >:= \int_a^b \omega(x) f(x) g(x) \delta x$$

Definition. The functions $f(x)$ and $g(x)$ are said to be *orthogonal* if $< f, g >= 0$.

Definition. A polynomial $p(x) = a_j x^j + a_{j-1} x^{j-1} + \cdots + a_1 x + a_0$ of degree j ($a_j \neq 0$) is said to be *normed* if $a_j = 1$.

Definition. Define $\Pi_j := \{p \mid x_j + a_{j-1} x_{j-1} + \cdots + a_0\}$ as the set of all normed polynomials of degree j.

Definition. Define $\overline{\Pi}_j := \{p \mid p(x) \text{ has degree} \leq j\}$ as the set of all polynomials with degree less than or equal to j, normed or not.

Theorem. For each $\omega(x)$, an infinite sequence $\{p_j\}_{j \geq 0}$ of normed polynomials exist, such that:

$$\begin{array}{lll} & p_j \in \Pi_j & j = 0, \ldots, +\infty \\ \text{and} & \forall i, j \;\; < p_i, p_j >= 0 & \text{if } i \neq j \end{array}$$

Theorem. (main theorem of Gaussian quadrature)
If $x_1, \ldots, x_n$ are the roots of the n-th orthogonal polynomial $p_n(x)$ respect to $\omega(x)$ in (a, b)

and if $\alpha_1, \ldots, \alpha_n$ is the solution of the system

$$\sum_{k=1}^{n} p_i(x_k) \alpha_k = \begin{cases} < p_0, p_0 > & \text{if } i = 0 \\ 0 & \text{if } i = 1, \ldots, n-1 \end{cases}$$

then

$$\begin{array}{lll} (i) & \forall k = 1, \ldots, n & \alpha_k > 0 \\ (ii) & \forall p \in \overline{\Pi}_{2n-1} & \int_a^b \omega(x) p(x) \delta x = \sum_{k=1}^{n} \alpha_k p(x_k) \end{array}$$

The expression $\sum_{k=1}^{n} \alpha_k p(x_k)$ is the Gaussian quadrature formula, and the weights and nodes are choosen as explained in the theorem. The following theorem informs about the error committed with these formulas.

Theorem. If f is C^{2n} in $[a, b]$, then

$[a,b]$	$\omega(x)$	Name
$[-1,+1]$	1	Gauss-Legendre
$[-1,+1]$	$(1-x^2)^{-\frac{1}{2}}$	Gauss-Chebyshev
$[0,+\infty)$	e^{-x}	Gauss-Laguerre
$(-\infty,+\infty)$	e^{-x^2}	Gauss-Hermite

Table 4: Gaussian integration rules.

n	α_k	x_k
1	2	0
2	1.00000 00000 00000	±0.57735 02691 89626
3	0.55555 55555 55556	±0.77459 66692 41483
	0.88888 88888 88889	0.00000 00000 00000
4	0.34785 48451 37454	±0.86113 63115 94053
	0.65214 51548 62546	±0.33998 10435 84856
5	0.23692 68850 56189	±0.90617 98459 38664
	0.47862 86704 99366	±0.53846 93101 05683
	1.56888 88888 88889	0.00000 00000 00000
6	0.17132 44923 79170	±0.93246 95142 03152
	0.36076 15730 48139	±0.66120 93864 66265
	0.46791 39345 72691	±0.23861 91860 83197

Table 5: Weights and nodes for the Gauss-Legendre quadrature formulas.

$$\int_a^b \omega(x)f(x)\delta x - \sum_{k=1}^{n} \alpha_k f(x_k) = \frac{f^{(2n)}(\xi)}{(2n)!} < p_n, p_n > \qquad \text{for some } \xi \text{ in } (a,b)$$

Gauss was the first to derive such a quadrature formula, and he centered on the special case $\omega(x) = 1$ and $[a,b] = [-1,+1]$. In this case, the orthogonal polynomials are defined by

$$p_i := \frac{i!}{(2i)!}\frac{\delta^i}{\delta x^i}(x^2-1)^i, \quad \text{for } i = 0, 1, \ldots$$

Up to a factor, these polynomials are the Legendre polynomials. Other authors had studied other cases for $\omega(x)$ and $[a,b]$, giving rise to other series of orthogonal polynomials and other quadrature formulas. In table 4 the most relevant are listed. The weights and nodes of the correspondent Gaussian quadrature formulas may be precomputed for several n and used directly; in table 5 the coefficients and weights are shown for the case of Gauss-Legendre quadrature.

Herein, no proofs have been given of any of the theorems enunciated; for such proofs, and further explanations of numerical integration methods, consult [Fro70] and [SB79].

D Approximation to linear opacity.

In this appendix it is shown the mathematical background that allows to approximate the expression of the opacity $\alpha = 1 - \exp(-\tau l)$, by simply $\alpha = \tau l$. Two different derivations will be exposed.

D.1 Serie development.

Taylor theorem : If $f^{(n)}(x)$ is continuous in $[a, b]$ and may be derived in (a, b), then two points ξ_1 and ξ_2 exist in (a, b), such that:

$$f(b) = f(a) + f'(a)(b-a) + f''(a)\frac{(b-a)^2}{2!} + \ldots + f^{(n)}(a)\frac{(b-a)^n}{n!} + R_n$$

where R_n is called the *remainder term*, and may be written as:

Lagrange remainder:

$$R_n = \frac{f^{(n+1)}(\xi_1)(b-a)^{n+1}}{(n+1)!}$$

Cauchy remainder:

$$R_n = \frac{f^{(n+1)}(\xi_2)(b-\xi_2)^n(b-a)}{n!}$$

The particular case of $a = 0$, it is known as McLaurin theorem.

McLaurin theorem : If $f^{(n)}(x)$ is continuous in $[0, a]$ and may be derived in $(0, a)$, then two points ξ_1 and ξ_2 exist in $(0, a)$, such that:

$$f(a) = f(0) - f'(0)a + f''(0)\frac{a^2}{2!} + \ldots + f^{(n)}(0)\frac{a^n}{n!} + R_n$$

where R_n may alternatively be written as $R_n = \frac{f^{(n+1)}(\xi_1)a^{n+1}}{(n+1)!}$ or $R_n = \frac{f^{(n+1)}(\xi_2)(a-\xi_2)^n a}{n!}$ [Spi69] [Ayr71].

The function $f(x) = e^{-x}$ is continuous in $\mathbf{R}$ and it is infinitely continuously derivable in $\mathbf{R}$. Applying McLaurin theorem to $f(x) = e^{-x}$ in the interval $[0, x]$, results:

$$\begin{aligned} e^{-x} &= 1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \ldots + R_n \\ &= 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \ldots + R_n \end{aligned}$$

Cutting the serie in its third term,

$$e^{-x} = 1 - x + R_1$$

with $R_1 = \frac{e^{-\xi}x^2}{2}$ for some ξ in $(0, x)$.

$$-e^{-x} = -1 + x - R_1$$

$$1 - e^{-x} = x - R_1$$

As $\xi > 0$, $e^{\xi} > 1$ and $0 < e^{-\xi} < 1$. Consequently,

$$R_1 = \frac{e^{-\xi}x^2}{2} < \frac{x^2}{2}$$

Approximating $1 - e^{-x}$ to x, the error committed is less than $\frac{x^2}{2}$ [13]. If $x < 1$, the quantity $\frac{x^2}{2}$ will be small, and it will be a good approximation. So, $1 - e^{-\tau l}$ may be satisfactorily approximatted to τl if $\tau l << 1$ (note that τl is always > 0).

D.2 Infinitesimal.

It is also possible to see the approximation by calculating a limit.

$$\lim_{x \to 0} \frac{1 - e^{-x}}{x} = \frac{0}{0}$$

Applying l'Hôpital rule,

$$\lim_{x \to 0} \frac{1 - e^{-x}}{x} = \lim_{x \to 0} \frac{e^{-x}}{1} = 1$$

That is, $1 - e^{-x}$ and x are equivalent infinitesimals, $1 - e^{-x} \sim x$. This means that, when x is sufficiently small, $1 - e^{-x}$ may be satisfactorily approximated to x. Consequently, $1 - e^{-\tau l}$ may be approximated to τl provided that $\tau l << 1$.

[13] It may also be used directly, that the numerical error committed when omitting the terms following the n-th, in an alternate convergent serie, is smaller than the absolute value of the first term omitted.

References

[AK90] J. Arvo and D. Kirk. The rendering equation. *Computer Graphics*, 24(4):63–66, August 1990.

[Arv93] J. Arvo. Transfer equations in global illumination. *Notes*, 1993.

[Ayr71] F. Ayres. *Cálculo diferencial e integral. Teoría y problemas.* McGraw Hill, 1971.

[BdE74] S. Burbano and de Ercilla. *Física General.* Editorial Librería General, Zaragoza 1974.

[BJN+94] P. Brunet, R. Juan, I. Navazo, A. Puig, J. Sole, and D. Tost. Modeling and visualiation trough data compression. *Data Visualization*, 1994.

[Bli82] J.F. Blinn. Ligth reflection functions for simulation of clouds and dusty surfaces. *Computer Graphics*, 16(3):21–29, July 1982.

[CCH95] L.W. Chang, H.W. Chen, and J.R. Ho. Rendering of surface and volume details in volume data. *Computer Graphics Forum*, 4(3):421–430, 1995.

[CFM+94] P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution volume dataset modeling and visualization based on simplicial complexes. *ACM Proceedings of 1994 Symposium on Volume Visualization*, pages 19–26, October 1994.

[Chr95] P. H. Christensen. Hierarchical techniques for global illumination. *PhD dissertation, University of Washington*, 1995.

[CHRU85] L.S. Chen, G.T. Herman, R.A. Reynolds, and J.K. Udupa. Surface shading in the cuberille environment. *IEEE Computer Graphics and Applications*, 5(12):33–43, December 1985.

[Com70] Physical Science Study Committee. *Física.* Editorial Reverté S.A., 1970.

[DCH88] R.A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *Computer Graphics*, 22(4):65–74, August 1988.

[FDFH93] J. Foley, A. Van Damm, S. Feiner, and J. Hughes. *Computer Graphics, Principles and Practice.* Addison Wesley, 1993.

[Fro70] C.E. Froberg. *Introducción al Análisis Numérico.* Vicens Universidad, 1970.

[Gla89] A. Glassner. *An Introduction to Ray Tracing.* Academic Press, 1989.

[Gla95] A. Glassner. *Principles of Digital Image Synthesis.* Morgan-Kaufman Series, 1995.

[Hal88] R. Hall. *Illumination and Color in Computer Generated Imaginery.* Springer Verlag, 1988.

[Kaj87] J.T. Kajiya. The rendering equation. *Computer Graphics*, 20(4):143–150, August 1987.

[Kau90] A. Kaufman. *Volume Visualization.* IEEE Computer Society Press, August 1990.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Computer Graphics*, 21(4):163–169, July 1987.

[Lev88] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8:29–37, May 1988.

[Lev90] M. Levoy. A hybrid ray tracer for rendering polygon and volume data. *IEEE Computer Graphics and Applications*, 10(8):33–40, March 1990.

[Max95] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.

[MHC90] N. Max, P. Hanrahan, and R. Crawfis. Area and volume coherence for efficient visualization of 3d scalar functions. *Computer Graphics, Proceedings of San Diego Workshop on Volume Visualization*, 24(5):27–33, November 1990.

[Mur93] S. Muraki. Volume data and wavelet transform. *IEEE Computer Graphics and Applications*, 13(4):50–56, July 1993.

[NA92] K. Novins and J. Arvo. Controlled precision volume integration. *1992 Workshop on Volume Visualization*, 1992.

[NAS92] K. Novins, J. Arvo, and D. Salesin. Adaptative error bracketing for controlled-precision volume rendering. *Technical Report TR92-1312, Cornel University Department of Computer Science*, 1992.

[NFHL91] G.M. Nielson, T.A. Foley, B. Hamman, and D. Lane. Visualizing and modeling scattered multivariate data. *IEEE Computer Graphics and Applications*, pages 47–55, May 1991.

[PD84] T. Porter and T. Duff. Compositing digital images. *ACM Computer Graphics*, 18(3):253–259, August 1984.

[Sab88] P. Sabella. A rendering algorithm for visualizing 3d scalar fields. *Computer Graphics*, 22(4):51–58, August 1988.

[SB79] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, 1979.

[Sea63] F.W. Sears. *Fundamentos de Física III.Óptica.* Aguilar S.A. Ediciones, 1963.

[Sob94] L. M. Sobierajski. Global illumination models for volume rendering. *PhD dissertation, State University of New York*, 1994.

[Spi69] M.R. Spiegel. *Cálculo superior. Teoría y problemas.* McGraw Hill, 1969.

[TPN95] D. Tost, A. Puig, and I. Navazo. A volume visualization algorithm using a coherent extended weigth matrix. *Computer and Graphics*, 19(1):37–45, 1995.

[UK88] C. Upson and M. Keeler. V-buffer: Visible volume rendering. *Computer Graphics*, 22:59–64, August 1988.

[WG91] J. Wilhems and A. Van Gelder. A coherent projection approach for direct volume rendering. *ACM Computer Graphics*, 25(4), July 1991.

[Wil91] J. Wilhems. Decisions in volume rendering. *Course Annotes of the SIGGRAPH Tutorial: State-of-the-Art in Volume Visualization*, 8, July 1991.

[WM92] P. Williams and N. Max. A volume density optical model. *1992 Workshop on Volume Visualization*, 1992.