Concepts for Volume Rendering

Jerry Tessendorf
Rhythm & Hues Studios

August 1, 2003

Contents

1 Applications **2**

2 Attributes **2**

3 Radiative Transfer Framework **4**
- 3.1 Radiance 4
- 3.2 Volume Scattering 4
- 3.3 Surface Reflection and Refraction 5
- 3.4 Light Sources 5
  - 3.4.1 Point Light 5
  - 3.4.2 Spot Light 5
  - 3.4.3 Area Light 6
  - 3.4.4 Volume Light 6
- 3.5 Radiative Transfer Equation 6
- 3.6 Generalize Radiative Transfer Equation 6

4 Traditional Volume Rendering $\leftrightarrow$ Single Scattering **8**
- 4.1 Pure Volume Rendering 9
  - 4.1.1 Numerical Evaluation 9

5 Multiple Scattering Preliminaries:
Path Integral Radiative Transfer **10**
- 5.1 Radiative Transfer Propagator 10
- 5.2 Path Integral for the Propagator 11

6 Multiple Scattering Approximations **13**
- 6.1 Alternative Schemes 14
- 6.2 Multiple Scatter 14
- 6.3 WKB Approximation 15
  - 6.3.1 Evaluating the Transform Path Integral 15
  - 6.3.2 The Primary Path Mip Mapping Argument 16
  - 6.3.3 Evaluating the Light Path Integral 17
  - 6.3.4 Ray Marching 17
  - 6.3.5 Primary Path and the Volume Blur Scale 18
  - 6.3.6 Further reduction 19
  - 6.3.7 Summary: L_{2+} in the End 20

7 Multiple Scattering Motion Blur **20**

8 Volume Renderer Functionality **20**

1 Applications

```
Clouds,
smoke,
water volume,
water splash,
atmospherics,
ghosts,
characters,
explosions,
dust,
weather, storms,
godrays, energy beams
```

2 Attributes

In radiative transfer and volume rendering, there are several attributes which are related to each other, but have different roles to play. Here is a small list of these attributes.

Density The most fundamental material attribute is the density, labelled ρ. In a physical cloud for example, it is the mass of the collection of water droplets within a unit volume, with units gram per cubic meter. In computer graphics problems, it is more abstract, but represents the portion of a given volume occupied by the material. By its nature, the density is a positive valued quantity (although FX TDs like to drive recklessly and make the density negative sometimes) and has no upper bound. This lack of upper bound is important for rendering cloudy phenomena. The other attributes discussed in this section are dependent on the density, as noted in each of them. The density ρ varies from point to point in the volume. In clouds, it can vary by as much as a factor of 1000.

Scattering Coefficient The volume material scatters light in all directions. The *scattering coefficient* b is the typical number of scattering events per unit length of the material. Typically b varies from point to point in the material along with the density, and is usually modelled as

$$b(\mathbf{x}) = \kappa_b \; \rho(\mathbf{x}) \tag{1}$$

The coefficient κ_b is constant valued throughout the volume. This value of κ_b however depends strongly on color. For example, the sky is blue because air molecules have larger values of κ_b in the blue than any other color. So it is important to keep the color dependence of scattering.

Scattering Phase Function When a scattering event happens, the light generally is not scattered uniformly in all directions. The *phase function* gives the angular distribution of scattered light. Ocean water, water clouds, and tissue have suprisingly similar phase functions in which a very large fraction of the scattered light is scattered close to the original direction of propagation, with some light scattered to wide angles. The phase function is labelled $P(\hat{\mathbf{n}}, \hat{\mathbf{n}}')$, which is the fractional density of light scattered from incoming angle $\hat{\mathbf{n}}$ to outgoing angle $\hat{\mathbf{n}}'$. In the cases of ocean water, water clouds, and tissue, the dependence on the incoming and outgoing directions is in the form $P(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')$, so that only the relative angle between the two directions is important. For (high altitude) ice clouds, the ice crystals frequently have special faceted shapes, and the air dynamics tends to align the crystals. In these circumstances, the phase function depends on more directions describing the incoming and outgoing directions with respect to the orientation of the crystal facets.

Absorption Coefficient In addition to scattering light, the volume material also absorbs light. This is described by an *absorption coefficient* a, which is related to the density by

$$a(\mathbf{x}) = \kappa_a \, \rho(\mathbf{x}) \tag{2}$$

The absorption coefficient a is roughly the number of absorption events per unit length of material. As with the scattering coefficient, the color dependence of this attribute can be important in some problems. For example, red sunsets are a consequence of the volumetric light attenuation of light coming from below the horizon. The blue-green color of water, and objects in water, is a consequence of the color character of both the absorption and scattering coefficients.

Extinction Coefficient The last of the coefficients is the *extinction* c, which is just the sum of the scattering and absorption:

$$c(\mathbf{x}) = a(\mathbf{x}) + b(\mathbf{x}) = \kappa_c \, \rho(\mathbf{x}) \tag{3}$$

and $\kappa_c = \kappa_a + \kappa_b$. In practice, it is frequenly useful to independently adjust all three values κ_a, κ_b, κ_c and not worry about these kinds of relationships.

Transmissivity and Opacity The various coefficients above cause a loss of light in the volume and a redistribution of the radiance throughout. Consequently the volume is opaque to some degree, depending on the path the light takes through the volume, the density distribution along the path, and the values of κ_a and κ_b, and the exact meaning of opacity one wishes to use for a particular application. Closely related to opacity is the transmissivity, which is simply $1-$opacity. Transmissivity is important because the solution of the radiative transfer problem – and volume rendering in general – includes references to transmissivity more naturally than opacity. To be explicit, for a path described as the set of points $\mathbf{x}(s), \; 0 <= s <= s_{max}$, the transmissivity associated with extinction is

$$T_{ext}(\{\mathbf{x}\}) = \exp\left\{ - \int_0^{s_{max}} ds \; c(\mathbf{x}(s)) \right\} \tag{4}$$

Again, because c is color dependent for some types of problems, T_{ext} is also color dependent.

The attribute list above contains items related to the optical behavior of a volume. Obviously, the spatial distribution of density is the most critical factor in making a volume appear to be a cloud, smoke, dust, etc. There are several methods of modeling the spatial arrangement of density, listed below:

Voxel Arrays This is the simplest conceptually and in terms of implementation. With a voxel array and ray marching a volume renderer is had, as long as you are happy with relatively low resolution because the memory footprint grows very fast. Voxel arrays can be used in conjunction with the other three methods below to dynamically generate voxel arrays in subregions of the total volume, as the ray marcher encounters them. This was done very effectively by Cinesite in the film SPHERE to render and fly through nebular interstellar regions, using Implicit Surfaces and noise functions as the driver for "voxelizing" any subregion when needed[9].

Particles Particles can be used to model density distributions in two general ways. In a macroscopic approach, the particles can be large objects that overlap, possibly combined with a displacement function to make them irregularly shaped. The volumetric density is nonzero inside the particles. The volume renderer must deal directly with the shape of the particles and how to compute a reasonable density value in regions of overlap of two or more particles. This is the approach Cinesite used for the Cerebro atmospheric look in XMEN 2: the particles were realized as three sided pyramids and the volume renderer computed the contributions from the insides of the these "tetrads"[10].

Alternatively, the particles may be microscopic, i.e. smaller than the resolution of the camera. The detailed shape of the particles is not so important as the tiny amount of screen space each one occupies. Rendering millions of such particles builds up the density as the fraction of a given volume occupied by particles. The *bamf* effect of Nightcrawler in XMEN 2 was created and rendered this way, using 2-10 million particles per frame. This method allows very flexible control via existing and future FX particle software, and extremely fine spatial detail. It also is efficient in terms of memory, using a constant amount of memory independent of the number of particles rendered. The memory footprint of the particle renderer at Cinesite was dominated by the size of the image plane because it was not tiled. For 2K frames, the memory footprint was around 120 MB[8].

Implicit Surfaces Implicit surfaces are useful for volume modeling because implicit surface equations exist throughout the volume of interest, and so can be used to drive the content of the interior of the surface. There is also a fairly large amount of academic literature on volume rendering implicit surfaces via modified ray marching (see for example [11]).

Procedural This representation of a volume is highly open and potentially the most time consuming rendering approach. The render proceeds with some form of ray marching, querying a function which returns the density $\rho(\mathbf{x})$ at any point $\mathbf{x}$ desired. This approach eliminates any potential efficiencies which the other methods above can exploit because there is no knowledge of how the volume is structured. On the other hand, this approach is also the most robust and flexible because it relies on very little knowledge of the volume.

3 Radiative Transfer Framework

In this section we discuss the formulation of the volume rendering problem in terms of radiative transfer of light through the medium and reflection/refraction at surfaces. In order to work practically with the radiative transfer formulation, the equation is generalized to a time-dependent version, where time is abstract and ultimately the final solution of interest is obtained in the long-time limit. This generalization is beneficial however because it allows us to sort the description of the light propagation into a series of paths through the scene. In the special cases of a vacuum ($\rho = 0$ everywhere) or weak single scattering only, these paths are implemented in a straight forward way as ray tracing or ray marching4. For more scattering events, the number of paths mushrooms and it is useful to apply multiple scatter approximations6.

3.1 Radiance

The fundamental quantity of light in radiative transfer is radiance. The radiance $L(\mathbf{x}, \hat{\mathbf{n}})$ is the "light density" (exactly, the power per square meter per steradian per nanometer) at the location $\mathbf{x}$ propagating in the direction $\hat{\mathbf{n}}$. If a pinhole camera is located at $\mathbf{x}_c$, pointed in the direction $\hat{\mathbf{n}}_c$, then at point (pixel) $\mathbf{p}$ on the image plane receives the light intensity

$$I(\mathbf{p}) = L\left(\mathbf{x}_c, \frac{\mathbf{p} - \hat{\mathbf{n}}_C}{\sqrt{1+p^2}}\right) \tag{5}$$

The reason for the minus sign in $(\mathbf{p} - \hat{\mathbf{n}}_C)/\sqrt{1+p^2}$ is that the angle in L is the direction of propagation, and $\hat{\mathbf{n}}_C$ is the viewing direction, which is exactly oppositely oriented.

3.2 Volume Scattering

Scattering of light from one direction into others is described by the scattering coefficient and the phase function. As light travels along a path, the about of light scattered per unit length is

$$b(\mathbf{x}) \int d\Omega' \; P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') \; L(\mathbf{x}, \hat{\mathbf{n}}') \tag{6}$$

This is similar in structure to the contribution of a surface light reflected or refracted from a surface using a BRDF. In fact:

3.3 Surface Reflection and Refraction

For a single surface (open or closed) described by a set of points $\mathbf{X}(\vec{\sigma})$ with a coordinate system $\vec{\sigma}$ on the surface, the light reflected/refracted is

$$\int d^2\sigma \sqrt{g(\vec{\sigma})}\, \delta\left(\mathbf{x} - \mathbf{X}(\vec{\sigma})\right) \int d\Omega'\, \mathcal{BRDF}(\vec{\sigma}, \hat{\mathbf{n}}, \hat{\mathbf{n}}')\, L(\mathbf{X}(\vec{\sigma}), \hat{\mathbf{n}}') \tag{7}$$

where the "area density" $g(\vec{\sigma})$ is the determinant of the metric tensor $\mathbf{g}$ of the surface[12]:

$$\mathbf{g}(\vec{\sigma}) = \sum_{i=1}^{3} \nabla_\sigma X_i(\vec{\sigma})\, \nabla_\sigma X_i(\vec{\sigma}) \tag{8}$$

For multiple independent surfaces $\mathbf{X}_a(\vec{\sigma}_a)$, $a = 1, \ldots, N$, the total contribution from surface interaction is

$$\sum_{a=1}^{N} \int d^2\sigma_a \sqrt{g_a(\vec{\sigma}_a)}\, \delta\left(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a)\right) \int d\Omega'\, \mathcal{BRDF}_a(\vec{\sigma}_a, \hat{\mathbf{n}}, \hat{\mathbf{n}}')\, L(\mathbf{X}_a(\vec{\sigma}_a), \hat{\mathbf{n}}') \tag{9}$$

3.4 Light Sources

Lights in general are a distribution of energy emitted from a collection of points in a variety of directions, labelled $S(\mathbf{x}, \hat{\mathbf{n}})$. Some examples are:

3.4.1 Point Light

A point light is a source that is located at a position $\mathbf{x}_S$, has no size, emits in all directions equally, and has total irradiance H. The mathematical form that fits this description is

$$S(\mathbf{x}, \hat{\mathbf{n}}) = \frac{H}{4\pi}\delta(\mathbf{x} - \mathbf{x}_S) \tag{10}$$

3.4.2 Spot Light

A spot light is located at a single position $\mathbf{x}_S$ like a point light, but emits preferentially in some directions more than others. It has the form

$$S(\mathbf{x}, \hat{\mathbf{n}}) = \frac{H}{4\pi}\delta(\mathbf{x} - \mathbf{x}_S)\, F(\hat{\mathbf{n}}) \tag{11}$$

The dimensionless angular distribution function F is normalized so that $\int d\Omega\, F(\hat{\mathbf{n}}) = 4\pi$. For example, a conically shaped spotlight with no penumbra or edge softening would have

$$F_{cone}(\hat{\mathbf{n}}) = \begin{cases} F_0 & \hat{\mathbf{n}} \cdot \hat{\mathbf{n}}_C > \cos(\theta_C) \\ 0 & \hat{\mathbf{n}} \cdot \hat{\mathbf{n}}_C \leq \cos(\theta_C) \end{cases} \tag{12}$$

and the constant is $F_0 = 2\pi\, (1 - \cos(\theta_C))$ for the cone half angle θ_C.

3.4.3 Area Light

An area light is analogous to a collection of spotlights distributed on a surface. Using the surface description from above, the area light has the form

$$S(\mathbf{x},\hat{\mathbf{n}}) = \frac{H}{4\pi}\int d^2\sigma \sqrt{g(\vec{\sigma})}\,\delta\left(\mathbf{x}-\mathbf{X}(\vec{\sigma})\right)\;F(\vec{\sigma},\hat{\mathbf{n}}) \tag{13}$$

This representation of an area light is actually a nice general way to represent area, spot, and point lights, since spot lights can be recovered by setting $F(\vec{\sigma},\hat{\mathbf{n}}) = \delta(\vec{\sigma}-\vec{\sigma}_S)F(\hat{\mathbf{n}})$, and point lights by further specializing to $F(\hat{\mathbf{n}}) = 1$.

3.4.4 Volume Light

$S(\mathbf{x},\hat{\mathbf{n}})$ is arbitrary.

3.5 Radiative Transfer Equation

Assembling the scattering, reflection/refraction, and attentuation, the light field throughout the volume satisfies the radiative transfer equation

$$\begin{aligned}\left\{\hat{\mathbf{n}}\cdot\nabla + c(\mathbf{x})\right\} L(\mathbf{x},\hat{\mathbf{n}}) &= b(\mathbf{x})\int d\Omega'\;P(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')\;L(\mathbf{x},\hat{\mathbf{n}}') \\ &+ \sum_{a=1}^{N}\int d^2\sigma_a\sqrt{g_a(\vec{\sigma}_a)}\,\delta\left(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a)\right)\int d\Omega'\;\mathcal{BRDF}_a(\vec{\sigma}_a,\hat{\mathbf{n}},\hat{\mathbf{n}}')\;L(\mathbf{X}_a(\vec{\sigma}_a),\hat{\mathbf{n}}') \\ &+ S(\mathbf{x},\hat{\mathbf{n}})\end{aligned} \tag{14}$$

3.6 Generalize Radiative Transfer Equation

Solving equation 14 analytically is pretty much impossible. Various numerical schemes have been used in a variety of applications. Some these schemes are discussed in section 6. It turns out that a lot of progress can be made in formulating a solution and applying ray tracing, path integrals, etc by augmenting the field with a kind of time dependence. In this case, the time is labeled s, and actually has units of length. This artifical time comes into the radiative transfer problem by changing equation 14 from a boundary value problem to an initial value problem, but the initial value is trivially zero. So we replace 14 with the equation

$$\begin{aligned}\left\{\frac{\partial}{\partial s} + \hat{\mathbf{n}}\cdot\nabla + c(\mathbf{x})\right\} L(s,\mathbf{x},\hat{\mathbf{n}}) &= b(\mathbf{x})\int d\Omega'\;P(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')\;L(s,\mathbf{x},\hat{\mathbf{n}}') \\ &+ \sum_{a=1}^{N}\int d^2\sigma_a\sqrt{g_a(\vec{\sigma}_a)}\,\delta\left(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a)\right)\int d\Omega'\;\mathcal{BRDF}_a(\vec{\sigma}_a,\hat{\mathbf{n}},\hat{\mathbf{n}}')\;L(s,\mathbf{X}_a(\vec{\sigma}_a),\hat{\mathbf{n}}') \\ &+ S(\mathbf{x},\hat{\mathbf{n}})\end{aligned} \tag{15}$$

and the initial condition $L(s=0,\mathbf{x},\hat{\mathbf{n}}) = 0$. The solution to the radiative transfer equation 14 is the long time limit of the solution of equation 15:

$$L(\mathbf{x},\hat{\mathbf{n}}) = L(s\to\infty,\mathbf{x},\hat{\mathbf{n}}) \tag{16}$$

As a first exercise in using this approach, lets remove volumetric material, so that only surfaces and lights remain. Also, we will use only lights that are point, spot, or area lights. The light source then has the general form

$$S(\mathbf{x},\hat{\mathbf{n}}) = \sum_{\mu}\frac{H_\mu}{4\pi}\int d^2\sigma_\mu\sqrt{g_\mu(\vec{\sigma}_\mu)}\,\delta\left(\mathbf{x}-\mathbf{X}_\mu(\vec{\sigma}_\mu)\right)\;F_\mu(\vec{\sigma}_\mu,\hat{\mathbf{n}}) \tag{17}$$

and the summation index μ runs over the set of lights in the scene. Setting $\rho = 0$, the generalized radiative transfer equation reduces to

$$\begin{aligned}\left\{\frac{\partial}{\partial s} + \hat{\mathbf{n}}\cdot\nabla\right\} L(s,\mathbf{x},\hat{\mathbf{n}}) &= \sum_{a=1}^{N} \int d^2\sigma_a \sqrt{g_a(\vec{\sigma}_a)}\, \delta\left(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a)\right) \int d\Omega'\, \mathcal{BRDF}_a(\vec{\sigma}_a, \hat{\mathbf{n}}, \hat{\mathbf{n}}')\, L(s, \mathbf{X}_a(\vec{\sigma}_a), \hat{\mathbf{n}}') \\ &+ \sum_{\mu} \frac{H_\mu}{4\pi} \int d^2\sigma_\mu \sqrt{g_\mu(\vec{\sigma}_\mu)}\, \delta\left(\mathbf{x} - \mathbf{X}_\mu(\vec{\sigma}_\mu)\right)\; F_\mu(\vec{\sigma}_\mu, \hat{\mathbf{n}}) \qquad (18)\end{aligned}$$

This has the nice result that we can turn it into an itegral equation:

$$\begin{aligned}L(s,\mathbf{x},\hat{\mathbf{n}}) &= \int_0^s ds' \sum_{a=1}^{N} \int d^2\sigma_a \sqrt{g_a(\vec{\sigma}_a)}\, \delta\left(\mathbf{x} - \hat{\mathbf{n}}(s-s') - \mathbf{X}_a(\vec{\sigma}_a)\right) \int d\Omega'\, \mathcal{BRDF}_a(\vec{\sigma}_a, \hat{\mathbf{n}}, \hat{\mathbf{n}}')\, L(s', \mathbf{X}_a(\vec{\sigma}_a), \hat{\mathbf{n}}') \\ &+ \int_0^s ds' \sum_{\mu} \frac{H_\mu}{4\pi} \int d^2\sigma_\mu \sqrt{g_\mu(\vec{\sigma}_\mu)}\, \delta\left(\mathbf{x} - \hat{\mathbf{n}}(s-s') - \mathbf{X}_\mu(\vec{\sigma}_\mu)\right)\; F_\mu(\vec{\sigma}_\mu, \hat{\mathbf{n}}) \qquad (19)\end{aligned}$$

Both of the terms on the right hand side have three dimensional delta functions of the form $\delta\left(\mathbf{x} - \hat{\mathbf{n}}(s-s') - \mathbf{X}\right)$, while we are integrating over the s' variable. The combination of the delta function with the integration picks out the unique (if any) value of s' which satisfies

$$\hat{\mathbf{n}}\cdot(\mathbf{x} - \mathbf{X}) = s - s' \qquad (20)$$

Keeping in mind that we are going to take $s \to \infty$, there is always a value of s' that will satisfy this as long as the left hand side is positive, meaning that $\mathbf{x}$ and $\mathbf{X}$ must be positioned relative to each other in general alignment with the propagation direction $\hat{\mathbf{n}}$. So the integration terms

$$\int_0^s ds'\, \delta\left(\mathbf{x} - \hat{\mathbf{n}}(s-s') - \mathbf{X}\right) \;\to\; \theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x}-\mathbf{X})\right)\, \delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x}-\mathbf{X})\right) \qquad (21)$$

The projection matrix $\Pi(\hat{\mathbf{n}}) = 1 - \hat{\mathbf{n}}\hat{\mathbf{n}}$ removes the portion of $\mathbf{x} - \mathbf{X}$ parallel to $\hat{\mathbf{n}}$, so the delta function restricts only the perpendicular behavior now. θ_H is the Heaviside step function, which is equal to 1 for a positive argument and 0 for a negative argument.

Putting these results in the integral equation 19 for the radiance, it has been reduced to

$$\begin{aligned}L(s,\mathbf{x},\hat{\mathbf{n}}) &= \sum_{a=1}^{N} \int d^2\sigma_a \sqrt{g_a(\vec{\sigma}_a)}\, \delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a))\right)\; \theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a))\right) \\ &\times \int d\Omega'\, \mathcal{BRDF}_a(\vec{\sigma}_a, \hat{\mathbf{n}}, \hat{\mathbf{n}}')\, L\left(s - \hat{\mathbf{n}}\cdot(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a)), \mathbf{X}_a(\vec{\sigma}_a), \hat{\mathbf{n}}'\right) \\ &+ \sum_{\mu} \frac{H_\mu}{4\pi} \int d^2\sigma_\mu \sqrt{g_\mu(\vec{\sigma}_\mu)}\, \delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x} - \mathbf{X}_\mu(\vec{\sigma}_\mu))\right)\; \theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x} - \mathbf{X}_\mu(\vec{\sigma}_\mu))\right) F_\mu(\vec{\sigma}_\mu, \hat{\mathbf{n}}) \qquad (22)\end{aligned}$$

As motivated earlier, we now take $s \to \infty$, and recover the classic rendering equation for surfaces in a "vaccuum":

$$\begin{aligned}L(\mathbf{x},\hat{\mathbf{n}}) &= \sum_{a=1}^{N} \int d^2\sigma_a \sqrt{g_a(\vec{\sigma}_a)}\, \delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a))\right)\; \theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x} - \mathbf{X}_a(\vec{\sigma}_a))\right) \\ &\times \int d\Omega'\, \mathcal{BRDF}_a(\vec{\sigma}_a, \hat{\mathbf{n}}, \hat{\mathbf{n}}')\, L(\mathbf{X}_a(\vec{\sigma}_a), \hat{\mathbf{n}}') \\ &+ \sum_{\mu} \frac{H_\mu}{4\pi} \int d^2\sigma_\mu \sqrt{g_\mu(\vec{\sigma}_\mu)}\, \delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x} - \mathbf{X}_\mu(\vec{\sigma}_\mu))\right)\; \theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x} - \mathbf{X}_\mu(\vec{\sigma}_\mu))\right) F_\mu(\vec{\sigma}_\mu, \hat{\mathbf{n}}) \qquad (23)\end{aligned}$$

4 Traditional Volume Rendering ↔ Single Scattering

Having established that classic surface rendering is achieved when the density of the volume material is zero, the next step is to look at the single scattering approximation. All volumetric renders available today are based on single scattering, although some modify the some of the intensity algorithms to be more like multiple scattering. This list includes the Houdini ray marcher being hosted by Caleb Howard, and the approach used in Wren. At one time Blue Sky talked about a multiple scatter renderer in the context of the short *Bunny*. However, in practice what was done was not volumetric (surface scattering only), and only computed up to the second bounce, which in volumetric language used here is the single scatter result (because the first incidence on surfaces is not counted).

The first step is to compute the unscattered light. This is not the same as the classic rendering problem in equations 18 through 23. Now we want to set only the scattering term to zero. This gives

$$\begin{aligned}\left\{\frac{\partial}{\partial s}+\hat{\mathbf{n}}\cdot\nabla+c(\mathbf{x})\right\}L(s,\mathbf{x},\hat{\mathbf{n}}) &= \sum_{a=1}^{N}\int d^2\sigma_a\sqrt{g_a(\vec{\sigma}_a)}\,\delta\left(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a)\right)\int d\Omega'\,\mathcal{BRDF}_a(\vec{\sigma}_a,\hat{\mathbf{n}},\hat{\mathbf{n}}')\;L(s,\mathbf{X}_a(\vec{\sigma}_a),\hat{\mathbf{n}}') \\ &+ S(\mathbf{x},\hat{\mathbf{n}}) \qquad (24)\end{aligned}$$

Just as with the zero density case, there is an integral equation equivalent to 24. Its form is identical with one additional factor inside the integrals:

$$\exp\left\{-\int_{s'}^{s}ds''\;c(\mathbf{x}-\hat{\mathbf{n}}(s-s''))\right\} \qquad (25)$$

Recalling the same procedure for evaluating the integral over s, the zero-scatter term is

$$\begin{aligned}L_0(\mathbf{x},\hat{\mathbf{n}}) &= \sum_{a=1}^{N}\int d^2\sigma_a\sqrt{g_a(\vec{\sigma}_a)}\,\delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a))\right)\;\theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a))\right) \\ &\times \exp\left\{-\int_0^{\hat{\mathbf{n}}\cdot(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a))}ds\;c(\mathbf{x}-\hat{\mathbf{n}}s)\right\}\;\int d\Omega'\,\mathcal{BRDF}_a(\vec{\sigma}_a,\hat{\mathbf{n}},\hat{\mathbf{n}}')\;L(\mathbf{X}_a(\vec{\sigma}_a),\hat{\mathbf{n}}') \\ &+ \int_0^\infty ds\;\;\exp\left\{-\int_0^s ds'\;c(\mathbf{x}-\hat{\mathbf{n}}s')\right\}\;S(\mathbf{x}-\hat{\mathbf{n}}s,\hat{\mathbf{n}}) \qquad (26)\end{aligned}$$

This exponential damping factor applied to the surface interactions is the fog term commonly used.

For lights that are not volume lights, the light term in equation 26 has form for each light

$$\begin{aligned}DSM(\mathbf{x},\hat{\mathbf{n}}) &= \frac{H}{4\pi}\int d^2\sigma\sqrt{g(\vec{\sigma})}\,\delta\left(\Pi(\hat{\mathbf{n}})\cdot(\mathbf{x}-\mathbf{X}(\vec{\sigma}))\right) \\ &\times \exp\left\{-\int_0^{\hat{\mathbf{n}}\cdot(\mathbf{x}-\mathbf{X}(\vec{\sigma}))}ds\;c(\mathbf{x}-\hat{\mathbf{n}}s)\right\} \\ &\times \theta_H\left(\hat{\mathbf{n}}\cdot(\mathbf{x}-\mathbf{X}(\vec{\sigma}))\right)F(\vec{\sigma},\hat{\mathbf{n}}) \qquad (27)\end{aligned}$$

This is the general form of a deep shadow map. If the extinction were zero, then the deep shadow map would have the translation symmetry $DSM(\mathbf{x}+\hat{\mathbf{n}}R,\hat{\mathbf{n}}) = DSM(\mathbf{x},\hat{\mathbf{n}})$ and could be reduced to just one two-dimensional map.

Now to the single scatter part: Starting with the zero scatter solution, we set the radiance L equal to the zero scatter radiance L_0 plus more L_1. The more satisfies keeping just the single scatter part of that, the radiative transfer equation reduces to

$$\begin{aligned}\left\{\frac{\partial}{\partial s}+\hat{\mathbf{n}}\cdot\nabla+c(\mathbf{x})\right\}L_1(s,\mathbf{x},\hat{\mathbf{n}}) &= b(\mathbf{x})\int d\Omega'\;P(\hat{\mathbf{n}},\hat{\mathbf{n}}')\;L_0(\mathbf{x},\hat{\mathbf{n}}') \\ &+ \sum_{a=1}^{N}\int d^2\sigma_a\sqrt{g_a(\vec{\sigma}_a)}\,\delta\left(\mathbf{x}-\mathbf{X}_a(\vec{\sigma}_a)\right) \\ &\times \int d\Omega'\,\mathcal{BRDF}_a(\vec{\sigma}_a,\hat{\mathbf{n}},\hat{\mathbf{n}}')\;L_1(s,\mathbf{X}_a(\vec{\sigma}_a),\hat{\mathbf{n}}') \qquad (28)\end{aligned}$$

4.1 Pure Volume Rendering

Concentrating for now on just the volumetric part (ignore surface contributions), the integral version of this equation is

$$L_1(s,\mathbf{x},\hat{\mathbf{n}}) = \int_0^s ds'\, T(s',s,\mathbf{x},\hat{\mathbf{n}})\, b(\mathbf{x}-\hat{\mathbf{n}}(s-s')) \int d\Omega'\, P(\hat{\mathbf{n}},\hat{\mathbf{n}}')\, DSM(\mathbf{x}-\hat{\mathbf{n}}(s-s'),\hat{\mathbf{n}}') \qquad (29)$$

In an effort to restore some cleanliness to these equations, the transmissivity has been reintroduced:

$$\begin{aligned} T(s',s,\mathbf{x},\hat{\mathbf{n}}) &\equiv \exp\left\{-\int_{s'}^{s} ds''\, c(\mathbf{x}-\hat{\mathbf{n}}(s-s''))\right\} \\ &= \exp\left\{-\int_0^{s-s'} ds''\, c(\mathbf{x}-\hat{\mathbf{n}}s'')\right\} \end{aligned} \qquad (30)$$

Taking the $s \to \infty$ limit, the final single scatter solution is

$$L_1(\mathbf{x},\hat{\mathbf{n}}) = \int_0^\infty ds\, T(0,s,\mathbf{x},\hat{\mathbf{n}})\, b(\mathbf{x}-\hat{\mathbf{n}}s) \int d\Omega'\, P(\hat{\mathbf{n}},\hat{\mathbf{n}}')\, DSM(\mathbf{x}-\hat{\mathbf{n}}s,\hat{\mathbf{n}}') \qquad (31)$$

This equation is the volume rendering equation for single scattered light for any type of lighting that does not include volumetric lights. The algorithm used by Wren and the Houdini ray marcher is exactly this algorithm, restricted further to lights that are point sources, e.g. point lights and spot lights. In those cases, the DSM reduces further and the volume rendering becomes

$$\begin{aligned} L_1(\mathbf{x},\hat{\mathbf{n}}) &= \int_0^\infty ds\, T(0,s,\mathbf{x},\hat{\mathbf{n}}) \\ &\times\; b(\mathbf{x}-\hat{\mathbf{n}}s)\, P\left(\hat{\mathbf{n}},\hat{\mathbf{N}}\right) \\ &\times\; T\left(0, |\mathbf{x}-\hat{\mathbf{n}}s-\mathbf{x}_L|, \mathbf{x}-\hat{\mathbf{n}}s, \hat{\mathbf{N}}\right) F(\hat{\mathbf{N}}) \end{aligned} \qquad (32)$$

where the unit vector $\hat{\mathbf{N}}$ is

$$\hat{\mathbf{N}} = \frac{\mathbf{x}-\mathbf{x}_L-\hat{\mathbf{n}}s}{|\mathbf{x}-\mathbf{x}_L-\hat{\mathbf{n}}s|} \qquad (33)$$

and $\mathbf{x}_L$ is the position of the light.

My limited study of the Wren volume rendering routine `evalVolume()` gave me the impression that equation 32 is being used, with the exception that the middle factor bP was not present, effectively setting that term to 1. In a sense, Wren's volume rendering corresponds to isotropic scattering (scattering uniformly in all directions) because of the lack of a phase function. Most likely phase functions can be added without substantial trouble to the existing software. More problematic, however, is the lack of the b factor, meaning that low density portions of the volume contribute more than their fair share. For example, a hole inside a cloud would have density $\rho = 0$ inside the hole, and $b = 0$ inside it as well. Consequently, there should not be a contribution of scattered light from that hole, as equation 32 properly computes. Wren's implementation however generates a contribution from that hole that is the same strength as regions outside the hole. This may or may not be modifiable within the existing volume rendering framework in Wren. One way it might possibly be modified is to change the definition of color to include a factor proportional to the density ρ (which is different from the quantity called density in `evalVolume` – that quantity is more like $1-T$). Similarly, the Houdini ray marcher has both of these limitations, and presumably is also able to be modified if desired.

4.1.1 Numerical Evaluation

Numerical implementation of equation 32 and similar algorithms requires integrating over the parameter s, which can be accomplished as a accumulation along the path $\mathbf{x} - \hat{\mathbf{n}}s$. The most straightforward way of doing this is to march with a fixed step Δs and add up the value of the integrand at each march point. The number of terms N in the sum can be related to visual quality issues, convergence issues, or resource constraints. The visual character of this approach was investigated in [2], where artifacts were noted for the simple direct summation. Recommendations in [2] were to jitter the exact evaluation point in order to break up the systematc character of the artifacts.

A second method of improvement is to employee more accurate standard techniques of numerical integration[13]. For example, both Romberg or Gauss Quadrature integration are known to be more accurate. It is possible that the net effect of better use of integration methods could include faster rendering and/or better rendered image quality.

5 Multiple Scattering Preliminaries: Path Integral Radiative Transfer

Multiple scattering volume rendering has always been a very hard problem to solve. The reason for this is that multiple scattering physically involves light travelling down many different paths that are unrelated to the lights or the camera before directly entering the camera. A physical cumulus cloud, for instance, scatters the light 100 times or more as the light traverses the interior of the cloud and finally leaves the cloud for our eyes and cameras. Thinking of this as a ray tracing problem, the number of ray paths in the cloud is exponentially large compared to single scattering volume rendering, and is much more compute intensive than the global illumination of surfaces. Hence, not only is multiple scattering hard, but we have to use an algorithm that is much faster than a ray trace in order to arrive at a practical algorithm.

There is a variety of numerical techniques in use to handle multiple scattering. All but one of them achieve practical levels of computational time and resources by limiting the amount of scattering they handle. This class of techniques is discussed below and a list of them provided. The one remaining algorithm available is very different from the others, and achieves its speed and efficiency by handling multiple scattering as a Level of Detail effect. Mutiple scattering causes three effects: spatial blurring, angular blurring, and attenuation. Most multiple scattering numerical procedures are time consuming because effectively they work to calculate these blur kernels. The WKB approximation is the only method that explicitly provides a formula for the blur kernels based on volume structure, and so reduces the multiple scatter rendering problem to that of only applying the blur kernels. This method is not only faster than the others, but can also be much faster than single scattering because the blur is used to accelerate ray marching.

The rest of this subsection is devoted showing the path integral expression that solves exactly the full volumetric radiative transfer problem, without surfaces. The path integral is essentially and infinite dimensional functional integral over all of the paths that light may travel through the scattering volume. The paths are not straight because scattering curves them.

5.1 Radiative Transfer Propagator

A useful concept and quantity is the propagator. This is independent of the initial conditions and light sources, and provide a full prescription for computing solutions to radiative transfer. For light propagating without nearby surfaces, the radiative transfer equation is

$$\left\{ \frac{\partial}{\partial s} + \hat{\mathbf{n}} \cdot \nabla + c(\mathbf{x}) \right\} L(s, \mathbf{x}, \hat{\mathbf{n}}) = b(\mathbf{x}) \int d\Omega' \; P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') \; L(s, \mathbf{x}, \hat{\mathbf{n}}') + S(\mathbf{x}, \hat{\mathbf{n}}) \tag{34}$$

The propagator is a special function G depending on $(s, \mathbf{x}, \hat{\mathbf{n}})$ and also on a set of points and directios $(\mathbf{x}', \hat{\mathbf{n}}')$ such that it satisfies equation the "homogeneous" equation

$$\left\{ \frac{\partial}{\partial s} + \hat{\mathbf{n}} \cdot \nabla + c(\mathbf{x}) \right\} G(s, \mathbf{x}, \hat{\mathbf{n}}; \mathbf{x}', \hat{\mathbf{n}}') = b(\mathbf{x}) \int d\Omega' \; P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') \; G(s, \mathbf{x}, \hat{\mathbf{n}}'; \mathbf{x}, \hat{\mathbf{n}}) \tag{35}$$

and also has the special initial condition

$$G(s=0,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}',\hat{\mathbf{n}}') = \delta(\mathbf{x}-\mathbf{x}')\ \ \delta(\hat{\mathbf{n}}-\hat{\mathbf{n}}') \tag{36}$$

For a ray of light that begins at $\mathbf{x}'$ propagating in direction $\hat{\mathbf{n}}'$, G is the intensity of that light ray when it arrives at $\mathbf{x}$ propagating in direction $\hat{\mathbf{n}}$, having traversed a (curved) path with a path length of s. Since there is no way for the ray to arrive at the final point without going through a path at least as long as $|\mathbf{x}-\mathbf{x}'|$, G is zero for $s < |\mathbf{x}-\mathbf{x}'|$. Because of this form for the propagator, the general solution to volume scattering is

$$L(\mathbf{x},\hat{\mathbf{n}}) = \int_0^\infty ds\ \int d^3x'\ \int d\Omega'\ \ G(s,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}',\hat{\mathbf{n}}')\ S(\mathbf{x}',\hat{\mathbf{n}}') \tag{37}$$

For area, spot, and point lights, this becomes explicitly:

$$L(\mathbf{x},\hat{\mathbf{n}}) = \sum_a \frac{H_a}{4\pi}\int_0^\infty ds\ \int d^2\sigma\sqrt{g_a}\ \int d\Omega'\ \ G(s,\mathbf{x},\hat{\mathbf{n}};\mathbf{X}_a(\vec{\sigma}_a),\hat{\mathbf{n}}')\ F_a(\vec{\sigma}_a,\hat{\mathbf{n}}') \tag{38}$$

and in particular if we are dealing only with spot and point lights,

$$L(\mathbf{x},\hat{\mathbf{n}}) = \sum_a \frac{H_a}{4\pi}\int_{|\mathbf{x}-\mathbf{x}_a|}^\infty ds\ \int d\Omega'\ \ G(s,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}_a,\hat{\mathbf{n}}')\ F_a(\hat{\mathbf{n}}') \tag{39}$$

For a number of mathematically worthwhile reasons, it is useful to separate G into two contributions: one for the unscattered light and one for the scattered light. This is accomplished by setting

$$G(s,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}',\hat{\mathbf{n}}') = \delta(\mathbf{x}-\mathbf{x}'-\hat{\mathbf{n}}s)\ \delta(\hat{\mathbf{n}}-\hat{\mathbf{n}}')\ T(0,s,\mathbf{x},\hat{\mathbf{n}})\ \ +\ \ \Delta G(s,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}',\hat{\mathbf{n}}') \tag{40}$$

The scattered portion, ΔG has the initial condition

$$\Delta G(s=0,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}',\hat{\mathbf{n}}') = 0 \tag{41}$$

Going back to the discussion at the beginning of this section, equations 37-39 show that the propagator acts on the light field as convolutions in space and direction. The scattering term ΔG turns this convolution into a blur in both space and direction. So numerical evaluation of the convolutions can be assisted by using mip-mapping techniques for both the spatial and direction degrees of freedom, assuming that we have a way of knowing the extent of the blurring. In subsequent sections the WKB approximation will be applied to create an expression for that blurring.

5.2 Path Integral for the Propagator

There is an explicit, exact, but mathematically formal expression for the scattered light propagator ΔG. Developed and applied to ocean and cloud optics over several years[3, 4, 6], it employs a technique called path integration. Introductory literature on path integration include [14] and [15]. Here is a basic motivation for this approach:

Light rays in a scattering environment do not follow straight lines - the path the light takes is curved under the influence of scattering and attenuation. In general, for a light ray that starts out at a particular point in a particular direction, it may arrive at another particular point and direction after taking a (potentially circuitous) curved route through the volume. Many different paths are possible, but each path has a different amount of attenuation. The final intensity of the ray when it reaches the destination is a sum over all possible paths of the attenuation for each path. In addition, the propagator also sorts the paths according to the length of the paths taken. So for a starting point and direction $\mathbf{x}',\ \hat{\mathbf{n}}'$, ending point and direction $\mathbf{x},\ \hat{\mathbf{n}}$, and path length s, the propagator is

$$\Delta G(s,\mathbf{x},\hat{\mathbf{n}};\mathbf{x}',\hat{\mathbf{n}}') \sim \sum_{paths} \exp\{-\tau(path)\} \tag{42}$$

The formal mathematics of path (or functional) integration turns this heuristic into a precise statement of the solution.

To begin, we need some way of describing a path through the volume. Here we introduce a unit vector $\hat{\beta}(s')$, defined for path lengths $s' \leq s$, and assign $\hat{\beta}(s')$ to be the unit vector that is tangent to the path at each point along the path. From this definition, it follows that the points of the path are the set

$$\mathbf{x}' + \int_0^{s'} ds'' \, \hat{\beta}(s'') \tag{43}$$

for all s' with $0 \leq s' \leq s$. Since we are only interesting in light paths that end up at position $\mathbf{x}$, we have to impose the constraint on these paths so

$$\mathbf{x} = \mathbf{x}' + \int_0^{s} ds' \, \hat{\beta}(s') \tag{44}$$

There is also the constraints that the direction of propagation begins at $\hat{\mathbf{n}}'$ and ends at $\hat{\mathbf{n}}$, which is written as

$$\hat{\beta}(0) = \hat{\mathbf{n}}' \tag{45}$$

$$\hat{\beta}(s) = \hat{\mathbf{n}} \tag{46}$$

One consequence of light propagating along a curved path is that the expression for the transmissivity has to be generalized to

$$T(s_1, s_2, \mathbf{x}|\hat{\beta}) = \exp\left\{ -\int_{s_1}^{s_2} ds' \, c\left(\mathbf{x} - \int_0^{s'} ds'' \, \hat{\beta}(s'') \right) \right\} \tag{47}$$

in order to allow for extinction losses along curved paths.

Now that there is a description for curved paths, we need a mechanism for summing over all possible paths. The set of all possible paths is obtained by view the path tangent $\hat{\beta}(s')$ at any given point s' on the path as a fluctuating variable, able to take on any unit vector value. The summation over all possible paths is achieved by integrating $\hat{\beta}$ over the unit sphere at each point s' on the path. So the notation

$$\sum_{paths} \tag{48}$$

becomes

$$\sum_{paths} \;\rightarrow\; \int \prod_{s'=0}^{s} d\Omega_\beta(s') \tag{49}$$

The product is achieved conceptually by dividing up the path into N discrete point separated by arclength $\Delta s = s/N$, then taking the limit $N \rightarrow \infty$. This expression in turn is reduced to a more compact notation

$$\int \prod_{s'=0}^{s} d\Omega_\beta(s') \;\rightarrow\; \int d\mu_\beta \tag{50}$$

where μ_β is the integration measure on the infinite dimensional space of paths. In addition to the introduction of the measure, the "boundary" conditions of equations 44-46 must be enforced. This is acheived by employing delta functions in the integration. Overall, the sum over paths measure becomes

$$\sum_{paths} \;\rightarrow\; \int d\mu_\beta \; \delta\left(\mathbf{x} - \mathbf{x}' - \int_0^{s} ds' \, \hat{\beta}(s') \right) \; \delta(\hat{\beta}(0) - \hat{\mathbf{n}}') \, \delta(\hat{\beta}(s) - \hat{\mathbf{n}}) \tag{51}$$

To achieve the closed form expression for the propagator ΔG, what remains to be done is to build an explicit expression for the transmissivity $\exp\{-\tau\}$ for any path.

The final expression for the path transmissivity uses a particular representation of the phase function. Rigorously, it depends on being able to express the phase function in terms of a Fourier transform representation:

$$P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') = \int d^2p\ \tilde{Z}(\mathbf{p})\ \exp\left\{i\mathbf{p}\cdot(\hat{\mathbf{n}}-\hat{\mathbf{n}}')\right\} \tag{52}$$

This transform property is true for an enormous range of physical situations in the environment. Also, by the time the WKB approximation has been completed in section 6.3, the final expression can be generalized to phase functions which don't have a Fourier decomposition. Using this decomposition, the expression for the transmissivity is

$$\begin{aligned} \exp\{-\tau(path)\} \quad &\rightarrow \quad T(0, s, \mathbf{x}|\hat{\beta}) \\ &\times \quad \int d\mu_p \exp\left(i\int_0^s ds'\ \mathbf{p}(s')\cdot\frac{d\hat{\beta}(s')}{ds'}\right) \\ &\times \quad \left\{\exp\left\{\int_0^s ds'\ b\left(\mathbf{x}-\int_0^{s'} ds''\ \hat{\beta}(s'')\right)\tilde{Z}(\mathbf{p}(s'))\right\}-1\right\} \end{aligned} \tag{53}$$

There is a remarkable change of variables that can be made to this expression which simplifies it substantially, but also assists in the numerical implementation of volume rendering for spatially varying density. If we define the dimensionless scattering variable

$$\ell'(s', \mathbf{x}'|\hat{\beta}) \equiv \int_0^{s'} ds''\ b\left(\mathbf{x}-\int_0^{s''} dt\ \hat{\beta}(t)\right) \tag{54}$$

then this variable effectively is a remapping of the arclength of a path into the number of scattering lengths the path tranverses, which depends on the arclength and on the portion of space the path goes through. This scattering variable is monotonic, i.e. $d\ell'/ds' \geq 0$ at all points in space, so making this change of variable makes sense mathematically. In terms of notation, we will leave the dependence on the path implicit and only write $\ell'(s')$. The transmissivity of a path becomes (with $\ell = \ell'(s'=s)$)

$$\exp\{-\tau(path)\} \rightarrow \exp\left\{-\ell\frac{\kappa_c}{\kappa_b}\right\}\int d\mu_p \exp\left(i\int_0^\ell d\ell'\ \mathbf{p}(\ell')\cdot\frac{d\hat{\beta}(\ell')}{d\ell'}\right)\left\{\exp\left\{\int_0^\ell d\ell'\ \tilde{Z}(\mathbf{p}(\ell'))\right\}-1\right\} \tag{55}$$

What makes this change of variables so remarkable is that the end result in equation 55 no longer depends on the spatial structure of the volume, except through the endpoint of integrations ℓ. This makes it possible to perform approximations like the WKB method below in a practical manner, and be assured that the approximation works for all spatial distributions of the the volume density.

Assembling everything, the final exact formal expression for the scatter portion of the propagator is

$$\begin{aligned} \Delta G(s, \mathbf{x}, \hat{\mathbf{n}}; \mathbf{x}', \hat{\mathbf{n}}') \quad &= \quad \int d\mu_\beta\ \int d\mu_p\ \delta\left(\mathbf{x}-\mathbf{x}'-\int_0^s ds'\ \hat{\beta}(s')\right)\ \delta(\hat{\beta}(0)-\hat{\mathbf{n}}')\ \delta(\hat{\beta}(s)-\hat{\mathbf{n}}) \\ &\times \quad \exp\left\{-\ell\frac{\kappa_c}{\kappa_b}\right\}\ \exp\left(i\int_0^\ell d\ell'\ \mathbf{p}(\ell')\cdot\frac{d\hat{\beta}(\ell')}{d\ell'}\right)\left\{\exp\left\{\int_0^\ell d\ell'\ \tilde{Z}(\mathbf{p}(\ell'))\right\}-1\right\} \end{aligned} \tag{56}$$

The goal of section 6 below is to convert this exact but formal solution of the radiative transfer equation into an approximate but practical solution that we can apply numerically in volume rendering.

6 Multiple Scattering Approximations

Before beginning to construction of a useful approximation to the propagator, section 6.1 briefly looks at several analytical and numberical schemes that have been used in the past to approximately solve the radiative transfer problem. In section 6.2 the exact formula for the multiple scattering propagator is applied to generalize the single scattering solution to the exact solution. The result forms the basis for the WKB approximation that follows in section 6.3.

6.1 Alternative Schemes

Small Angle Approximation

Photon Density Diffusion

Photon Mapping

Discrete Ordinates

Legrendre Expansion

Monte Carlo Raytracing

Path Integral Raytracing

6.2 Multiple Scatter

The path integral results give us an approach to pursue multiple scattering. In this subsection this expression for the propagator is used to write an explicit exact formal expression for the radiance distribution in the volume. This is still within the context of ignoring surfaces.

Going back to equation 28, eliminating the surface term on the right hand side, but restoring the scattering term that was thrown out because it wasn't needed for single scattering, the radiative transfer equation is

$$\begin{aligned} \left\{ \frac{\partial}{\partial s} + \hat{\mathbf{n}} \cdot \nabla + c(\mathbf{x}) \right\} L(s, \mathbf{x}, \hat{\mathbf{n}}) &= b(\mathbf{x}) \int d\Omega' \; P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') \; L(s, \mathbf{x}, \hat{\mathbf{n}}') \\ &+ b(\mathbf{x}) \int d\Omega' \; P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') \; L_0(\mathbf{x}, \hat{\mathbf{n}}') \end{aligned} \tag{57}$$

Applying the propagator to solve this equation for the long-time solution

$$\begin{aligned} L(\mathbf{x}, \hat{\mathbf{n}}) &= L_0(\mathbf{x}, \hat{\mathbf{n}}) + \int_0^\infty ds \int d\Omega' \int d^3x' \; G(s, \mathbf{x}, \hat{\mathbf{n}}; \mathbf{x}', \hat{\mathbf{n}}') \, b(\mathbf{x}') \int d\Omega'' \; P(\hat{\mathbf{n}}', \hat{\mathbf{n}}'') \; L_0(\mathbf{x}', \hat{\mathbf{n}}'') \\ &= L_0(\mathbf{x}, \hat{\mathbf{n}}) + \int_0^\infty ds \, T(0, s, \mathbf{x}, \hat{\mathbf{n}}) \, b(\mathbf{x} - \hat{\mathbf{n}}s) \int d\Omega' \; P(\hat{\mathbf{n}}, \hat{\mathbf{n}}') \; L_0(\mathbf{x} - \hat{\mathbf{n}}s, \hat{\mathbf{n}}') \\ &+ \int_0^\infty ds \int d\Omega' \int d^3x' \; \Delta G(s, \mathbf{x}, \hat{\mathbf{n}}; \mathbf{x}', \hat{\mathbf{n}}') \, b(\mathbf{x}') \int d\Omega'' \; P(\hat{\mathbf{n}}', \hat{\mathbf{n}}'') \; L_0(\mathbf{x}', \hat{\mathbf{n}}'') \end{aligned} \tag{58}$$

The middle term in this expression is the single scattered radiance L_1. The last term is the multiple scattered radiance, and is without approximation at theis point. This term will be labeled L_{2+}. Using the path integral solution for ΔG this last term becomes

$$\begin{aligned} L_{2+}(\mathbf{x}, \hat{\mathbf{n}}) &= \int_0^\infty d\ell \; \exp\{-\ell/\omega_0\} \int d\mu_\beta \, d\mu_p \, \delta(\hat{\beta}(s) - \hat{\mathbf{n}}) \\ &\times \exp\left(i \int_0^\ell d\ell' \; \mathbf{p}(\ell') \cdot \frac{d\hat{\beta}(\ell')}{d\ell'} \right) \left\{ \exp\left\{ \int_0^\ell d\ell' \; \tilde{Z}(\mathbf{p}(\ell')) \right\} - 1 \right\} \\ &\times \int d\Omega' \; P(\hat{\beta}(0), \hat{\mathbf{n}}') \; L_0 \left(\mathbf{x} - \int_0^s ds' \; \hat{\beta}(s'), \; \hat{\mathbf{n}}' \right) \end{aligned} \tag{59}$$

This expression for L_{2+} is actually not very nasty. The infinite dimensional path integrals in it will be taken care of in section 6.3, so they are not worrisome factors. Computationally, the remaining sources of heavy work are the two integrals. The integral over ℓ is the ray marching process, written as a march over the number of scattering lengths instead of over path length. The effect of this rephrasing of the ray marching will be discussed in section 7.

The remaining integral over solid angle is no more effort than the one encountered in the single scatter term L_1. Since most light sources are spatially limited or directionally limited, this solid angle integration not as numerically intensive as one might think. Indeed, this term is at the heart of most any volume renderer. This is remarkable, because it says that, following treatment of the path integral, the total contribution of all scattering events beyond single scattering is no more difficult to compute that the single scattering term itself.

So if you can tolerate the computational resources to render single scattering, roughly doubling your resources will get you the complete global illumination solution of volume rendering.

6.3 WKB Approximation

Now that we have the exact but formal expression 59 for the multiple scattered light, the path integrals must be evaluated approximately in order to achieve some practical algorithms. There has been a great deal of progress in recent years on this portion of the problem, largely undocumented. Here we run through all of the major points.

6.3.1 Evaluating the Transform Path Integral

The first step is an approximate evaluation of the path integral over the transform variable $\mathbf{p}$. The path integral of interest is

$$\int d\mu_p \, \exp\left(i \int_0^{\ell} d\ell' \, \mathbf{p}(\ell') \cdot \frac{d\hat{\beta}(\ell')}{d\ell'} \right) \left\{ \exp\left\{ \int_0^{\ell} d\ell' \, \tilde{Z}(\mathbf{p}(\ell')) \right\} - 1 \right\} \tag{60}$$

The Fourier Transformed phase function $Z(\mathbf{p})$ is the critical part here in building the approximation. Multiple scattering tends to scramble information about the phase function. This in some ways justifies choosing Z to better fit approximation schemes, as along as the chosen Z still has some basis in reality. In fact, many natural settings have phase functions which are sharply peaked in the forward scattering direction, and it is common in the small angle approximation and others to use a gaussian phase function, for which Z is gaussian also, of the form

$$\tilde{Z}(\mathbf{p}) = \exp\left\{ -\frac{\mu}{2} p^2 \right\} \tag{61}$$

and the parameter μ is the square of the angular width of the phase function. For example, for deep ocean optics situations, $\mu \approx 0.01$. With μ acting as a small parameter, the dominant contribution to the path integral comes by making the series of approximations:

$$\begin{aligned}
\exp\left\{ \int_0^{\ell} d\ell' \, \tilde{Z}(\mathbf{p}(\ell')) \right\} - 1 \;\; &\rightarrow \;\; \exp\left\{ \int_0^{\ell} d\ell' \, \left(1 - \frac{\mu}{2} p^2(\ell') \right) \right\} - 1 \\
&\rightarrow \;\; e^{\ell} \left\{ 1 - \frac{\mu}{2} \int_0^{\ell} d\ell' \, p^2(\ell') \right\} - 1 \\
&\rightarrow \;\; \left(e^{\ell} - 1 \right) \left\{ 1 - \frac{\mu}{2\left(1 - e^{-\ell}\right)} \int_0^{\ell} d\ell' \, p^2(\ell') \right\} \\
&\rightarrow \;\; \left(e^{\ell} - 1 \right) \, \exp\left\{ -\frac{\mu}{2\left(1 - e^{-\ell}\right)} \int_0^{\ell} d\ell' \, p^2(\ell') \right\}
\end{aligned} \tag{62}$$

There are three reasons why this set of approximate manipulations work well:

- The physics of the problem generally puts μ at a small value, as previously discussed, justifying using it as a perturbation parameter;
- The outcome is a gaussian in $\mathbf{p}$, which allows further manipulation, as is done below;

- Looking at the behavior of $e^{\int \tilde{Z}} - 1$ in the $\mathbf{p} \to 0$ and $\mathbf{p} \to \infty$ limits, the approximate expression has exactly the same behavior.

With the outcome of these approximations being gaussian, the path integral of equation 60 can be evaluated to:

$$\left(e^{\ell} - 1\right) \exp\left\{ -\frac{\ell}{2\mu_\ell} \int_0^\ell d\ell' \left| \frac{d\hat{\beta}(\ell')}{d\ell'} \right|^2 \right\} \tag{63}$$

and μ_ℓ has been defined as

$$\mu_\ell = \mu \, \frac{\ell}{(1 - e^{-\ell})} \tag{64}$$

So with the aid of the approximations on the tranform path integral the expression for L_{2+} is

$$\begin{aligned} L_{2+}(\mathbf{x}, \hat{\mathbf{n}}) &= \int_0^\infty d\ell \, \exp\{-\ell/\omega_0\} \, \left(e^{\ell} - 1\right) \\ &\times \int d\mu_\beta \, \delta(\hat{\beta}(s) - \hat{\mathbf{n}}) \, \exp\left\{ -\frac{\ell}{2\mu_\ell} \int_0^\ell d\ell' \left| \frac{d\hat{\beta}(\ell')}{d\ell'} \right|^2 \right\} \\ &\times \int d\Omega' \, P(\hat{\beta}(0), \hat{\mathbf{n}}') \, L_0\left(\mathbf{x} - \int_0^s ds' \, \hat{\beta}(s'), \, \hat{\mathbf{n}}' \right) \end{aligned} \tag{65}$$

6.3.2 The Primary Path Mip Mapping Argument

The one part of the expression for L_{2+} that poses any complication is the term

$$L_0\left(\mathbf{x} - \int_0^s ds' \, \hat{\beta}(s'), \, \hat{\mathbf{n}}' \right) \tag{66}$$

because of the presence of the integral over $\hat{\beta}$ in its arguments. In fact, if this integral were absent, the problem would be solved, because in section 6.3.3 an exact evaluation is shown for the remaining path integral (which works when the L_0 factor does not contribute). Ultimately, what happens in this section is we find an argument for separating the L_0 from the path integral. So here goes....

The path integral sums up the contributions of many varied paths through the volume. However, each path is weighted by the factor

$$\exp\left\{ -\frac{1}{2\mu_\ell} \int_0^\ell d\ell' \left| \frac{d\hat{\beta}(\ell')}{d\ell'} \right|^2 \right\} \tag{67}$$

Consequently, there is a "primary path" that contributes the most, along with a set of "nearby paths" that contribute substantially also. Within that set of primary and nearby paths, summing up a term like

$$L_0\left(\mathbf{x} - \int_0^s ds' \, \hat{\beta}(s'), \, \hat{\mathbf{n}}' \right) \tag{68}$$

has the effect of spatially blurring it, centered around the primary path. The amount of spatial blurring depends on the contribution of the nearby paths, which in turn is a function of the distance of propagation and the optical properties. The formula for the amount of spatial blurring is given in section 6.3.5.

The approach we take in this approximation is to replace L_0 with a spatially blurred version of itself, evaluated along the primary path. In order to distinguish this version, we use the label

$$L_0\left(\mathbf{x} - \int_0^s ds' \, \hat{\beta}(s'), \, \hat{\mathbf{n}}' \right) \;\to\; L_{blur}\left(s, \mathbf{x} - \int_0^s ds' \, \hat{\beta}_0(s'), \, \hat{\mathbf{n}}' \right) \tag{69}$$

L_{blur} has an argument s, which tracks the spatial scale of the blurring. The path $\hat{\beta}_0(\ell')$ is the primary path, also presented in section 6.3.5.

Having used a little hand-waving in this section to set up a blurred source L_{blur} that depends on the primary path but not on the fluctuating paths, we are free to evaluate the remain path integral in a nice elegant way.

6.3.3 Evaluating the Light Path Integral

Sometimes things go your way, as is the case with the path integral that remains. Even though it has a gaussian form, it is complicated by the fact that $\hat{\beta}$ is a unit vector at every point on the path. Fortunately, an exact solution is available[16], expressed in terms of spherical harmonics:

$$\int d\mu_\beta \, \delta(\hat{\beta}(s)-\hat{\mathbf{n}}) \, \delta(\hat{\beta}(0)-\hat{\mathbf{n}}') \, \exp\left\{ -\frac{1}{2\mu_\ell} \int_0^\ell d\ell' \left| \frac{d\hat{\beta}(\ell')}{d\ell'} \right|^2 \right\} = \sum_{j=0}^{\infty} \exp\left\{ -\frac{\mu_\ell}{2} j(j+1) \right\} \sum_{m=-j}^{j} Y_{jm}(\hat{\mathbf{n}}) \, Y^*_{jm}(\hat{\mathbf{n}}') \tag{70}$$

Consequently, the radiance distribution is

$$\begin{aligned} L_{2+}(\mathbf{x}, \hat{\mathbf{n}}) &= \int_0^\infty d\ell \, \exp\{-\ell/\omega_0\} \left(e^\ell - 1 \right) \\ &\times \sum_{j=0}^{\infty} \exp\left\{ -\frac{\mu_\ell}{2} j(j+1) \right\} \sum_{m=-j}^{j} Y_{jm}(\hat{\mathbf{n}}) \\ &\times \int d\Omega_0 \, Y^*_{jm}(\hat{\mathbf{n}}_0) \int d\Omega' \, P(\hat{\mathbf{n}}_0, \hat{\mathbf{n}}') \, L_{blur}\left(s, \mathbf{x} - \int_0^s ds' \, \hat{\beta}_0(s'), \, \hat{\mathbf{n}}' \right) \end{aligned} \tag{71}$$

and the mapping between ℓ and s is

$$\ell = \int_0^s ds' \, b\left(\mathbf{x} - \int_0^{s'} dt \, \hat{\beta}_0(t) \right) \tag{72}$$

There are more reductions to be made to equation 71, but first we make a diversion to understand some aspects of both equations 71 and 72.

6.3.4 Ray Marching

Equation 71 contains a remarkable result that has not been used in any other approach to radiative transfer and volume rendering. The expression uses the dimensionless scattering length parameter ℓ as the independent integration variable. However, the definition of ℓ comes from equation 72, apparently making ℓ dependent on s (the original integration variable), $\mathbf{x}$, and the primary path of propagation. What has happened is that we have switched the roles of what is dependent and independent. Now ℓ is independent, and the pathlength s of the ray march into the volume is dependent on ℓ, $\mathbf{x}$, and the primary path. This switch has created a great simpifying situation in equation 71: every quantity in the equation is completely independent of the spatial distribution of the volume material, except for the last one, L_{blur}. In most any volume rendering schemes, the spatial structure of the volume is a factor of every portion of the renderer. Here, it has been isolated to one term coupled with a prescription for ray marching (i.e. equation 72). And the exponential nature of the factors in the integrand assure rapid, perhaps uniform convergence of the integration when marched in even steps of ℓ. So this approach not only simplifies the dependence on the volume structure, it also provides more robust convergence and accuracy behavior. I like it.

Now that there is a focus on the transformation between ℓ and s, we need to determine how to practically implement it. The first step toward a workable algorithm is recognizing that ℓ is quasi-monotonic in s, which means that if $s_1 > s_2$, then $\ell_1 \geq \ell_2$, and also if $\ell_1 > \ell_2$ then $s_1 > s_2$. This allows us to implement the following steps:

1. Maintain a table of $\{\ell, s\}$ pairs that are computed.

2. For the largest value ℓ in the table, the next step in the march, $\ell + \Delta\ell$ takes an additional pathlength Δs, which from equation 72 is

$$\Delta\ell = \int_s^{s+\Delta s} ds'\, b\left(\mathbf{x} - \int_0^{s'} dt\, \hat{\beta}_0(t)\right) \tag{73}$$

3. Use a strategy to find the value of Δs that satisfies this. There are many strategies possible.

4. Push this new pair, $\{\ell + \Delta\ell, s + \Delta s\}$ into the table.

5. Evaluate the appropriate integration terms.

Remember, up to this point, only *two* approximations have been made to arrive at a complete global illumination solution for the radiative transfer equation: (a) the phase function for multiple scattering was simplified to include primarily the forward scattering portion, and (b) the integration over paths of L_0 we assumed would spatially smooth L_0, which decoupled it from the actual path integral.

These two approximations are quite valid for many realistic natural circumstances, and also are not a hinderance at all for visual effects applications.

6.3.5 Primary Path and the Volume Blur Scale

There are additional simplifications and approximations that are worthwhile making to reduce the computational effect of this solution. First, it will be helpful to look at two quantities that have not been shown: the exact primary path, and the blur size for L_{blur}.

Primary Path

The primary path is found by minimizing the "energy functional"

$$\int_0^{\ell} d\ell' \left|\frac{d\hat{\beta}(\ell')}{d\ell'}\right|^2 \tag{74}$$

with respect to the independent components of $\hat{\beta}$. Since $\hat{\beta}$ is a unit vector, we can define spherical coordinates such that

$$\hat{\beta}_0(\ell') = (\sin\theta(\ell')\ \cos\phi(\ell'),\ \sin\theta(\ell')\ \sin\phi(\ell'),\ \cos\theta(\ell')) \tag{75}$$

With spherical coordinates, the energy function becomes

$$\int_0^{\ell} d\ell' \left\{\left(\frac{d\theta(\ell')}{d\ell'}\right)^2 + \left(\frac{d\phi(\ell')}{d\ell'}\right)^2 \sin^2\theta(\ell')\right\} \tag{76}$$

Since the angles θ and ϕ are independent and unconstrained, the minimization can take place on them simultaneously. The exact solution is

$$\cos\theta(\ell') = \sin\alpha\ \sin(D(\ell - \ell_0)) \tag{77}$$

$$\phi(\ell') = \phi_0 + \arctan\{\cos\alpha\tan(D(\ell' - \ell_0))\} + \arctan\{\cos\alpha\tan(D\ell_0)\} \tag{78}$$

and the integration constants ϕ_0, D, ℓ_0, and α are determined by the boundary conditions on the path.

Blur scale

I am not going to derive the quantity in these notes. There is a derivation in a different context, but same mathematics, in [5]. The result is that the blur should smooth out all structure smaller than

$$s\sqrt{24\mu_\ell} \tag{79}$$

Note that because of the dependence on s, the amount of blur is also dependent on the particular path taken into the volume, as it should be. In practice, the way to implement this is via 3D mip mapping: precompute the blurred data at a variety of blur levels (using less data for more blur when possible), then during the ray march interpolate across the precomputed set of blur scales to achieve the ones of interest.

6.3.6 Further reduction

There is one final observation that reduces the complexity of the multiple scattering problem without severe compromise. The blur scale in equation 79 is of the same size as the amount of deviation of the primary path from a straight line. That it, the differences between the two paths

$$\mathbf{x} - \int_0^{s'} dt\ \hat{\beta}(t) \tag{80}$$

and

$$\mathbf{x} - \hat{\mathbf{n}}\ s' \tag{81}$$

is a distance approximately or less than the blur scale. Consequently, marching along a straightline path will not be much in error compared to the primary path. So it is reasonable to replace equations 71, 72, and 73 with ones evaluated on a straightline:

$$\begin{aligned} L_{2+}(\mathbf{x},\hat{\mathbf{n}}) &= \int_0^\infty d\ell\ \exp\{-\ell/\omega_0\}\ \left(e^\ell - 1\right) \\ &\times\ \sum_{j=0}^\infty \exp\left\{-\frac{\mu_\ell}{2}j(j+1)\right\}\ \sum_{m=-j}^{j} Y_{jm}(\hat{\mathbf{n}}) \\ &\times\ \int d\Omega_0\ Y^*_{jm}(\hat{\mathbf{n}}_0)\int d\Omega'\ P(\hat{\mathbf{n}}_0,\hat{\mathbf{n}}')\ L_{blur}\left(s,\mathbf{x}-\hat{\mathbf{n}}\ s,\ \hat{\mathbf{n}}'\right) && (82) \\ \ell &= \int_0^s ds'\ b\left(\mathbf{x}-\hat{\mathbf{n}}\ s'\right) && (83) \\ \Delta\ell &= \int_s^{s+\Delta s} ds'\ b\left(\mathbf{x}-\hat{\mathbf{n}}\ s'\right) && (84) \end{aligned}$$

This reduction in turn generates more. Because the spherical harmonics are a complete orthogonal basis, without any loss of generality the phase function can be expanded as

$$P(\hat{\mathbf{n}},\hat{\mathbf{n}}') = \sum_{j=0}^\infty \sum_{m=-j}^{j}\ P_{jm}\ Y_{jm}(\hat{\mathbf{n}})\ Y^*_{jm}(\hat{\mathbf{n}}') \tag{85}$$

The solid angle integrations then reduce to

$$\int d\Omega_0\ Y^*_{jm}(\hat{\mathbf{n}}_0)\int d\Omega'\ P(\hat{\mathbf{n}}_0,\hat{\mathbf{n}}')\ L_{blur}\left(s,\mathbf{x}-\hat{\mathbf{n}}\ s,\ \hat{\mathbf{n}}'\right) = P_{jm}\ L^{jm}_{blur}(s,\mathbf{x},\hat{\mathbf{n}}) \tag{86}$$

where the last term is

$$L^{jm}_{blur}(s,\mathbf{x},\hat{\mathbf{n}}) = \int d\Omega'\ Y^*_{jm}(\hat{\mathbf{n}}')\ L_{blur}\left(s,\mathbf{x}-\hat{\mathbf{n}}\ s,\ \hat{\mathbf{n}}'\right) \tag{87}$$

6.3.7 Summary: L_{2+} in the End

Summarizing the development of last two sections, the radiance due to all contributions beyond single scattering are approximated by:

$$L_{2+}(\mathbf{x}, \hat{\mathbf{n}}) = \int_0^\infty d\ell \ \exp\{-\ell/\omega_0\} \ \left(e^\ell - 1\right) \sum_{j=0}^{\infty} \exp\left\{-\frac{\mu_\ell}{2} j(j+1)\right\} \sum_{m=-j}^{j} P_{jm} \, Y_{jm}(\hat{\mathbf{n}}) \, L^{jm}_{blur}\left(s, \mathbf{x}, \hat{\mathbf{n}}\right) \quad (88)$$

and s is the implicit solution of the equation

$$\ell = \int_0^s ds' \, b\left(\mathbf{x} - \hat{\mathbf{n}} \, s'\right) \quad (89)$$

7 Multiple Scattering Motion Blur

To implement this multiple scattering in a practical volume rendering code, there is one additional feature that must be discussed: motion blur. Here we outline the mathematical issues and indicate how to do it for the case of camera motion, but leave the details for later notes on an actual implementation.

For camera motion, the camera position $\mathbf{x}$ becomes a function of time $\mathbf{x}(t)$, and the view vector for any pixel $\hat{\mathbf{n}}$ also becomes a function of time $\hat{\mathbf{n}}(t)$. Motion blur averages the radiance over an exposure time T. Looking over equation 88, the only part affected by this averaging are the factors

$$Y_{jm}(\hat{\mathbf{n}}) \, L^{jm}_{blur}\left(s, \mathbf{x}, \hat{\mathbf{n}}\right) \ \rightarrow \ \frac{1}{T} \int_0^T dt \, Y_{jm}(\hat{\mathbf{n}}(t)) \, L^{jm}_{blur}\left(s(t), \mathbf{x}(t), \hat{\mathbf{n}}(t)\right) \quad (90)$$

8 Volume Renderer Functionality

Basic requirements for a standalone renderer

- Single scatter and multiple scatter options
- Can optionally call out single scatter followed by multiple scatter
- Spectral transmissivity
- Pervasive shading/expression language based on C++
- Mip-mapped volume rendering via section 6 WKB approach implemented as in section 7.
- Object occlusion via depth/alpha maps. (object rendering would be better but not essential)
- Embeddable architecture (into houdini, maya, voodoo, perl, commandline....)

Global Illumination requirements

- Triangulated beam tracing
- Basic surface object handling/rendering (for object occlusion).

References

[1] Henrik Wann Jensen, ”Global Illumination using Photon Maps”, *Rendering Techniques ’96*, (1996).

[2] Mark Pauly, Thomas Kollig, and Alexander Keller, ”Metropolis Light Transport for Participating Media”

[3] J. Tessendorf and D. Wasson, ”Scattering in the 3D Cloud Scene Simulator”, Arete Associates document, Feb 15, 1994.

[4] J. Tessendorf, ”Comparison between data and small-angle approximations for the in-water solar radiance distribution,” J. Opt. Soc. Am., **A5**, p. 1410-1418, (1988).

[5] J. Tessendorf, ”The underwater solar light field: analytical model from a WKB evaluation,” *Underwater Imaging, Photography, and Visibility*, SPIE Vol. 1537, (1991).

[6] J. Tessendorf, ”Radiative Transfer as a Sum Over Paths”, Phys. Rev. A, **35**, p. 872-878, (1987).

[7] W.S. Helliwell, ”Finite-difference solution to the radiative-transfer equation for in-water radiance”, J. Opt. Soc. Am. **A2**, p. 1325-1330, (1985).

[8] Jerry Tessendorf, Efficiently Rendering Gobs and Gobs of Particles, Cinesite Technical Notes, 2001

[9] Gokhan Kisacikoglu, The Making of Black-Hole and Nebula Clouds for the Motion Picture “Sphere” with Volumetric Rendering and the F-Rep of Solids, Siggraph 98 Sketches.

[10] Jerry Tessendorf, Bill LaBarge, Vijoy Gaddipati, Tetrad Volume and Particle Rendering in “X2”, Siggraph 2003 Sketches and Applications.

[11] B.T. Stander, J.C. Hart. A Lipschitz method for accelerated volume rendering. Proceedings of the 1994 Symposium on Volume Visualization, Oct. 1994. pp. 107-114.

[12] Millman and Parker, *Elements of Differential Geometry*, Prentice-Hall, (1977).

[13] Press, Teukolsky, Vetterling, and Flannery, *Numerical Recipes in C*, second edition, Cambridge University Press, (1992)

[14] Richard Feynman, *Statistical Mechanics: A Set of Lectures*, Perseus Publishing; 2nd edition (January 1998).

[15] R. P. Feynmann and A.R. Hibbs, *Quantum Mechanics and Path Integrals*,McGraw-Hill Higher Education, (June 1, 1965)

[16] H. Kleinert, “Path Integral on Spherical Surfaces in D Dimensions and on Group Spaces”, (1989).

[17] “Spherical Harmonics”