Computation of Global Illumination in a Participating Medium by Monte Carlo Simulation

S.N.Pattanaik and S.P.Mudur
Graphics & CAD Division,
National Centre for Software Technology,
Juhu, Bombay 400 049, INDIA

Abstract

This paper discusses techniques for the computation of global illumination in environments with participating medium using a Monte Carlo simulation of the particle model of light. Efficient algorithms and data structures for tracking the particles inside the volume have been developed. The necessary equation for computing the illumination along any given direction has been derived for rendering a scene with participating medium. A major issue in any Monte Carlo simulation is the uncertainty in the final simulation results. Various steps of the algorithm have been analysed to identify major sources of uncertainty. To reduce the uncertainty, suitable modifications to the simulation algorithm have been suggested using variance reduction methods of *forced collision*, *absorption suppression* and *particle divergence*. Some sample scenes showing the results of applying these methods are also included.

CR Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism; I.3.3 [Computer Graphics]: Picture/Image Generation.
General Terms: Algorithms
Additional Keywords and Phrases: Monte Carlo Simulation, Global Illumination, Participating Medium, Uncertainty Reduction.

1 Introduction

The Monte Carlo simulation technique has been used for a long time for the solution of problems in particle transport [1, 2]. In an earlier paper [3] the authors have described in detail its application to the computation of global illumination in a geometrically and optically complex 3-D environment. This paper is primarily concerned with the application of the Monte Carlo simulation technique for the computation of global illumination in the presence of a participating medium. Normally in illumination computations, we often make the assumption that the radiance of the light in a given direction is independent of position along that direction and that the medium through which the light passes does not in any way affect the radiance. In other words the medium does not interact with the light. Strictly speaking this is true only when light travels in vacuum. Also when light travels a short distance in a rare medium free of dust or smoke we ignore the presence of the medium as its effect does not make any visible difference to the illumination. However, when the above environmental conditions do not hold, the radiance along any direction changes with the position along that direction. The medium is said to be participating. The events taking place in the medium that are responsible for the change in the radiance

are: absorption, emission and scattering. Absorption results in a loss in the light energy and hence results in the reduction of the radiance. Scattering results in a change in the light propagation direction and hence results in the reduction of the radiance along the given direction. However it must be noted here that the radiance may also be augmented by the light scattered from other directions. Emission by the medium generates light energy and hence always has the effect of augmenting the radiance.

Pioneering work on the computation of illumination in the presence of participating (absorbing and scattering) medium for use in image synthesis was first carried out by Blinn [4]. For certain specific conditions, such as a uniformly scattering medium with a low scattering albedo and a single light source, Blinn formulated a simple equation for computing the radiance of light reaching the eye. A number of other researchers [5, 6, 7] have extended Blinn's work to accommodate more complex participating behaviours of the medium and to use ray tracing for image synthesis. All this work is however based on the use of a local illumination model in which the interaction of all the volumes and surfaces with one another with respect to the propagation of light is not accurately modeled. Rushmeier and Torrance [8] extended the radiosity method to include surface-volume, volume-surface and volume-volume interaction and computed global illumination by solving a set of linear equations one for each surface patch and for each volume element.

In this paper we present various details of the MonteCarlo simulation technique to compute global illumination in complex 3-D environments with participating media. In the particle model of light, particles denoting (*photons*) quanta of light energy, are emitted and propagated from the emitting surface and volume elements. These particles interact with the environment by getting scattered in the volume elements or may reach the surfaces and get reflected or transmitted by the surfaces. The scattered particles are responsible for the volume illumination and the reflected or transmitted particles are responsible for the surface illumination. These scattered, reflected and transmitted particles continue propagating in the environment till they are absorbed by a surface or volume. Global illumination is thus the result of these complex interactions of the particles with the volumes and surfaces. Computation of global illumination then amounts to the evaluation of the scattered particle flux at the volume elements and of the reflected and transmitted particle flux at the surface elements.

For simulating this process a finite number of particles are generated originating at different positions on the emitter surface or in the emitter volume, and are assigned different directions of propagation. For each particle the nearest surface along the propagation path is computed. In the absence of any participating volume the particle moves unhindered and hits the nearest surface along its path. At that surface the particle is either absorbed, reflected or transmitted. In the case of the particle traveling through a participating medium an additional step is needed to account for the interaction of the particle with the intervening volume. Depending on the interaction behaviour of the medium a suitable position for the particle-volume interaction is computed along the path of the particle. If the chosen position lies before the nearest surface along the particle path then the particle does not reach the surface. Instead it is either absorbed or scattered randomly in the volume at the computed position. For each particle the process is continued until the particle is absorbed or the particle leaks out from the bounds of the environment of interest. In the simulation discussed above at various stages suitable choices are made by Monte Carlo sampling of the respective probability distribution functions (*pdf*s). The simulation algorithm along with details of the volume interaction related computations are discussed in the next section.

Illumination computation by simulating the propagation of light starting from the light

sources in not entirely new. Progressive radiosity[9] is one of the most well established methods in which light energy is recursively shot from emitting surfaces and other bright surface to all other visible surfaces in the environment. Bidirectional ray tracing [10] is another method, a two pass method, in which in the first pass rays carry energy to points arranged in a 3D regular grid covering the environment. In the second pass ray tracing is carried out to compute the illumination of the visible points by collecting local contribution from the visible light sources and the visible 3D grid points where the light energy is accumulated in the first pass. There is however a fundamental difference between these and the Monte Carlo simulation method. Both in progressive radiosity and bidirectional ray tracing the process of shooting light is completely deterministic. Whereas in the Monte Carlo simulation it is non-deterministic and is potentially capable of simulating more general behaviour.

2 The Simulation Algorithm

For each particle repeat steps (1) to (4) below:

1. Choose the particle's wavelength by sampling the cumulative emission spectrum.
   In the presence of multiple light sources choose the emitter from which the particle will originate by sampling the emitter strength distribution at the chosen wavelength.
   Choose the position on the emitter at which the particle originates by position sampling the emitter surface geometry or the emitter volume.

2. Update the outgoing particle flux at the emitter.

3. Choose the direction in which the particle is emitted by sampling the directional emission distribution function.

4. Repeat steps (a) and (b) below until the particle is absorbed.

   (a) Choose the point of interaction.

   (b) If *the point of interaction is inside a volume*
   then /* Particle interacts with the volume.*/

   Sample the scattering/absorption albedo distribution to decide on the type of interaction.
   If the interaction type is *scattering* then

   i. Update the outgoing flux of the volume.

   ii. Assign scatter direction by sampling the directional scattering distribution function or phase function.

   else if *the point of interaction is on a surface*
   then /* Particle interacts with the surface.*/

   Sample the reflection/absorption albedo distribution to decide on the type of interaction.
   If the interaction type is *reflection* then

   i. Update the outgoing particle flux on the reflecting surface.

   ii. Assign reflection direction by sampling the surface bidirectional reflection distribution function.

It is clear that only after a sufficiently large number of particles have been tracked we can expect illumination computation of reasonable accuracy. The actual number of particles would depend on the geometry and optical complexity of the 3D environment. An important point to note is the truly progressive nature of this algorithm. Approximate illumination values start becoming available from quite early in the simulation process. And as the simulation progresses and more particles are tracked, the computed illumination keeps getting continuously refined. In principle it is therefore possible to carry on the simulation until the computed illumination remains invariant, within tolerance of course. In our present implementation however we carry out a simulation with the number of particles chosen apriory by the user depending on the environmental complexity. For example Fig.1a shows an environment consisting of 661 surfaces, small and large, which we have tracked using a million particles. So far we have found that in absence of a participating medium one million is adequate for such an environment and gives us highly satisfactory results. Also with the computing power available today tracking of a million particles is not at all prohibitive. For the given environment the average number of reflections that a particle undergoes is about 3.57 i.e. about three and a half million rays have been traced. By using a suitable acceleration technique[11] this ray tracing can be generally contained. In fact we have found that with the spatial subdivision based acceleration technique that has been implemented the average time for the Monte Carlo simulation method is more or less independent of the geometric complexity in the environment. The total time for the simulation was 1:03 hrs on DRS6000 (a 33MHZ SPARC/RISC processor). Fig.1b shows the same environment filled with non-absorbing, isotropically scattering gray medium with a scattering coefficient of 0.1. For this 10 million particles has been traced. The number of particles required in the presence of participating medium however depends on several factors : the optical thickness of the volume, the locality of the medium and the total number of volume elements. For the scene in Fig.1b the whole environment was assumed to be embedded inside the volume with a total of 15625 small volume elements. The total time for simulation was 27:24hrs.

2.1 Computing Flux Density

The outgoing flux is recorded at each surface patch and volume element by simply keeping a count of the outgoing particles from that patch during emission or reflection. This count is the direct estimator of the equilibrium particle flux density of the surface patch/volume element. The relation is as follows:

$$Outgoing\ Flux = \frac{Number\ of\ Particles\ leaving\ the\ element}{Total\ Number\ of\ Simulation\ Particles} * Total\ Source\ Strength$$

From this the flux density is computed as: $Flux\ Density = \frac{Total\ outgoing\ Flux}{Surface\ Area}$.
In the process of simulation a region under shadow will have very few particles while another region directly facing a light source will have a very large number of particles reaching that region. The positional distribution of the outgoing particles directly gives us the variation of brightness over the surface and volume. One method of capturing the distribution of particles is to have a predefined mesh structure associated with the receiver surface and volume. However in such a case, the choice of the mesh is important for eventually it is this which determines how well the illumination gradient has been captured. The automatic discretisation problem is being researched extensively[12, 13, 14] and any of these methods could be used. It is however important to note that in the Monte Carlo simulation the mesh structure only stores simulation results and plays no role in the actual simulation

process which is carried out by dealing with the surfaces/volume and reflectance/scattering behaviour without any simplifications or approximations. A further point to be noted is that in the Monte Carlo simulation, computation time depends only on the number of particles and the environmental complexity and is independent of the mesh structure. In our present implementation the particle distribution is accumulated over a rectangular mesh (in UV space) imposed over each surface and over a uniform spatial mesh imposed over the volume.

2.2 Assigning Positions to Emitted Particles

There are mainly three important characteristics of the emitter that influence the illumination [15]. They are *geometry* (emitter surface shape), *spectral distribution* (emitter strength at different wavelengths) and *luminous radiance distribution* (emitter strength in different directions at different positions).

Traditionally in computer graphics the light source is assumed to be a point or a line and if more realistic appearance is called for then area sources are simulated by a large collection of points or lines. The very first treatment of real area geometry has been in the radiosity method [16]. In the radiosity method any large or complex surface must be first broken down into small patches with constant emission strength before its use in illumination computation. On the other hand for Monte Carlo simulation the essence lies in devising the proper sampling strategy so that the particles are positionally distributed corresponding to the emitter strength variation over the surface and assigned path directions corresponding to the directional distribution of the emission strength. Sampling equations for a few typical surface geometries have been derived and tabulated in Table 1. These assume uniform emitter strength over the entire surface. If emission strength is not uniform over the surface of the emitter then position sampling has to account for this variation as well. Assuming that just like the position the emission strength variation is described as a function of two parameters *(u,v)* we first subdivide the surface into subpatches of nearly uniform emission strength. We then carry out discrete sampling among these subpatches based on their relative strengths and then sample for the exact position within the subpatch using the sampling equation for that surface.

2.3 Assigning Directions to Emitted Particles

Having assigned a position we must also be able to assign a direction to each emitted particle. A direction is defined by a pair (ϕ, θ) where ϕ is the circumferential angle and θ is the cone angle. It is generally assumed that the emission does not depend on the circumferential angle ϕ and hence radiance distribution is specified only along the cone angle by goniometric diagrams [15]. In such a case ϕ can be sampled as $\phi = 2\pi\xi_\phi$ where ξ_ϕ is a uniform random number chosen from the range 0 and 1 and θ is sampled by *Rejection Sampling* of the Goniometer Curve. For more specific cases like fixed direction and diffuse emission from an emitting surface and isotropic emission from the emitting volume the particle flight direction can be directly assigned as described below.

Fixed directional emission is only along one direction and the generated particle is assigned directly the only predefined direction associated with the emitter surface. This may be used for modeling sun light coming through the window pane.

In *diffuse emission* the emitted radiance is uniform in all directions and it can be shown that the generated particle must be assigned the direction given by the pair $(2\pi\xi_\phi, sin^{-1}\sqrt{\xi_\theta})$

where ξ_ϕ and ξ_θ are uniform random variables in the range 0 to 1.

Similarly for *isotropic emission* from a volume the particle direction is given by the pair $(2\pi\xi_\phi, cos^{-1}(1-2\xi_\theta))$.

2.4 Assigning Directions to Reflected Particles

A reflection direction is assigned by sampling the *brdf* of the surface. The *brdf* is usually defined as a function of a single variable θ_i, the angle of incidence. This is the angle which the incoming particle direction makes with the normal to the surface at the point of reflection. In the general case the rejection sampling method can be used. More specific cases like purely diffuse, purely specular and Phong model reflection are described below. For diffuse reflectance the reflected radiance distribution is uniform around the surface similar to the diffuse emission process discussed earlier. Hence the direction for a reflected particle is: $(2\pi\xi_\phi, sin^{-1}\sqrt{\xi_\theta})$ where ξ_ϕ and ξ_θ are uniform random variables in the range 0 to 1. For mirror reflectance the direction choice is simpler as there is only one direction to consider for a given incident direction. So at the position where the particle hits the receiver surface the reflection direction is computed from the incident direction and the normal to the surface at that point. If the surface *brdf* is given by Phong's specular model then the sampled reflection direction is given by the vector which makes angle $(cos^{-1}(1-r_1)^{\frac{1}{n+1}}, 2\pi r_2)$ with the mirror reflection of the incident vector[17]. Similarly for the surface *brdf* modelled using Ward's anisotropic reflectance model[18] the sampled direction is given by:

$$\begin{aligned} \delta &= \left[\frac{-\log \xi_1}{\cos^2\phi/\alpha_x^2 + \sin^2\phi/\alpha_y^2}\right] \\ \phi &= \tan^{-1}\left[\frac{\alpha_y}{\alpha_x}\tan(2\pi\xi_2)\right] \end{aligned}$$

where $(\delta,\ \phi)$ represent the direction of the bisector vector of the incident and sampled outgoing direction,
α_x, α_y are the standard deviation of the surface slope in the **x** and **y** direction.

3 Interaction in Absorbing and Scattering Medium

As the light energy passes through an absorbing and/or scattering medium the radiance is reduced by the interaction of the particles with the medium. The reduction in radiance on traveling a distance dS depends on i_λ, the starting radiance and K_λ, the extinction coefficient of the medium, and is given by:

$$di_\lambda = -K_\lambda i_\lambda dS$$

The change in radiance over a distance of length S from the point of origin of the light is arrived at by integrating the above equation in the range 0 to S. On integration we get:

$$i_\lambda(S) = i_\lambda(0)e^{-\int_0^S K_\lambda(u)du}$$

where $i_\lambda(0)$ and $i_\lambda(S)$ are respectively the radiance at the emitter and the radiance at a distance S away from the emitter for any given wavelength λ and u is the variable for integration which takes value in the range of 0 to S. This equation is known as Bouguer's Law. In the above equation $K_\lambda(u)$, the extinction coefficient at the wavelength λ, with units

of $length^{-1}$, is a physical property of the medium quantifying the interacting capability of the medium with light. This property has two parts: absorption and scattering and hence can be written as:

$$K_\lambda \;=\; a_\lambda \;+\; \sigma_\lambda$$

The relative contribution of each part to the scattering coefficient is given by its albedo. For example: scattering albedo (Ω) is given by $\frac{\sigma_\lambda}{K_\lambda}$.

The power term in Bouguer's equation $\int_0^S K_\lambda(u)du$ is termed *opacity* or *optical thickness*. Opacity is a measure of the ability of a given pathlength S of the medium to attenuate the light energy of a given wavelength λ. A large opacity means large attenuation. Opacity is a dimensionless parameter. From Bouguer's law the radiance of light on traveling a pathlength S inside a participating medium with extinction coefficient *K(u)* reduces to a factor of $e^{-\int_0^S K(u)du}$ of the original radiance. This factor may be interpreted as the probability of any particle traveling a path length S without undergoing interaction. Thus the probability of the particle interacting before traveling a pathlength S, is $1 - e^{-\int_0^S K(u)du}$. Since this probability is the cumulative probability of the particle interaction at every point along the path from 0 to S we get the following expression for the cumulative distribution function, *cdf*.

$$cdf \;=\; 1 \;-\; e^{-\int_0^S K(u)du}$$

By the principle of Random Variable Transformation [19], sampling of a *pdf* is carried out by finding an expression for its *cdf* and solving the equation $cdf = \xi$, where ξ is the uniform random number in the range 0 to 1. So

$$\xi \;=\; 1 \;-\; e^{-\int_0^S K(u)du} \quad or \quad 1-\xi \;=\; e^{-\int_0^S K(u)du}$$

For ξ uniformly distributed over the range 0 to 1, (1-ξ)=ξ_1 is also uniformly distributed over the same range 0 to 1. So the sampling equation is

$$\xi_1 \;=\; e^{-\int_0^S K(u)du} \quad or \quad log\xi_1 \;=\; -\int_0^S K(u)du = \; -Opacity$$

For a homogeneous medium the *Opacity* is $K \cdot S$ and hence path length sampling can be carried out conveniently by drawing a uniform random number (ξ_1) and computing the path length, S, from the equation $S = -\frac{log\xi_1}{K}$. However, for a medium which is not homogeneous in its participating properties, the sampling of path length requires the evaluation of the integral and hence is difficult. Howell [20] has proposed a solution to a similar problem by making the simplifying assumption that the interacting volume may be divided into plane increments of ΔS inside which the interaction properties are fairly homogeneous. Under this assumption the integration reduces to a summation as follows:

$$log\xi_1 \;=\; -\sum_{j=1}^{p} K_j \Delta S_j \quad or \quad log\xi_1 \;+\; \sum_{j=1}^{p} K_j \Delta S_j \;=\; 0$$

where K_j and ΔS_j are respectively the extinction coefficient and the pathlength in the j_th incremental volume. Now to find the path length, one has to incrementally trace the plane increments and check for the satisfaction of the inequality $log\xi_1 \;+\; \sum_{j=1}^{p} K_j \Delta S_j \;\geq\; 0$. The

first incremental slab p for which the inequality is satisfied contains the sampled point of interaction and the exact point of interaction or the path length S is given by

$$log\xi_1 \ + \ \sum_{j=1}^{p-1} K_j \Delta S_j + K_p \Delta S = 0$$

$$\Delta S = - \left(log\xi_1 \ + \ \sum_{j=1}^{p-1} K_j \Delta S_j \right) / K_p$$

$$S = \sum_{j=1}^{p-1} \Delta S_j + \Delta S = \sum_{j=1}^{p-1} \Delta S_j - (\sum_{j=1}^{p-1} K_j \Delta S_j + log\xi_1)/K_p$$

Though the exact method as stated above is not directly suitable for sampling in a complex 3D environment, a slight variation of this method makes it ideal for use. In this variation it may be assumed that the volume bounding the environment can be uniformly partitioned into small *voxel*s inside each of which the medium is fairly homogeneous. A particle traveling through the volume can be tracked through a list of *voxel*s very simply by using the 3D-DDA algorithm[21] and the above equations can be solved to determine the point of interaction. The pseudo code for this method is given below:

cumulative_pathlength=0
pathlength_measure = $log\xi_1$
for each voxel *along the particle path do*

if(pathlength_measure+$K_{voxel}\Delta S_{voxel} > 0$*)*
interaction will take place in this voxel.
S = *cumulative_pathlength - pathlength_measure/*K_{voxel}
Stop.

else
pathlength_measure = *pathlength_measure+*$K_{voxel}\Delta S_{voxel}$
cumulative_pathlength = *cumulative_pathlength+*ΔS_{voxel}

If control reaches here it means that the particle did not interact in the intervening volume.

4 Implementation Strategy

The volume structure assumed for the implementation of this technique is very similar to the one used in a ray tracing acceleration method for computing ray-surface interaction, namely the *Spatial Enumeration technique*[11]. Since finding the nearest surface along a particle path is done by performing ray-surface intersections, the same data-structure may be adopted. However, the requirement that the volume interaction properties within a *voxel* must be uniform would most often imply a fine subdivision of the environment. Such fineness if used, for both the acceleration of ray-surface intersection computations and the computation of volume interaction pathlength, will result in heavy memory overheads. If we consider the fact that in most of the environments the participating volume may be highly localised, for example: fire and smoke in a corner of a room, then fine subdivision of the entire environment is not necessary. We have adopted a two-level uniform volume

subdivision technique – a coarse subdivision into *cell*s for acceleration of the ray-surface intersection and a further subdivision of cells into *voxel*s. A preprocessor does the following:

- associates with each cell a list of surfaces and a list of participating volume elements whose bounding extents intersect the *cell*, and
- subdivides those cells with a nonempty list of volume elements, into *voxel*s. Each *voxel* is assigned just enough memory to capture the particle events during the simulation.

At the time of particle tracking both the *surface interaction pathlength* and *volume interaction pathlength* are computed simultaneously which as the reader will notice is a slight variation in the previous algorithm just described. The exact computation steps are given below:

1. *cumulative_pathlength*=0; *pathlength_measure* = $\log\xi_1$
2. Carry out a 3D-DDA on the *cell* structure and get the *cell* list ordered along the particle path.
3. For each *cell* do steps (a) to (c) below.
   (a) Compute the list of intersecting surfaces and find the the nearest point of surface intersection within the *cell* if any.
   (b) If the *cell* has a nonempty volume element list then carry out 3D-DDA on the fine *voxel* structure within the *cell* up to the farthest end of the *cell* along the particle path or up to the nearest surface intersection point whichever is nearer and generate the *voxel* list ordered along the particle path.
   (c) for each *voxel* in the list do the following

   if ($pathlength\_measure + K_{voxel}\Delta S_{voxel} > 0$)
   then /* volume interaction point reached.*/
   $volume\_interaction\_pathlength =$
   $cumulative\_pathlength$ - $pathlength\_measure / K_{voxel}$
   break
   else
   $pathlength\_measure = pathlength\_measure + K_{voxel}\Delta S_{voxel}$
   $cumulative\_pathlength = cumulative\_pathlength + \Delta S_{voxel}$

   (d) If either the nearest surface of intersection is found or the volume interaction point is reached then go to step (4).
4. Sample interaction distribution function to decide on the interaction type.

5 Modeling Participating Volumes

As can be seen in the above algorithm the participating volume model must be such that for each volume element we are able to do the following:

- Cell-Volume Classification : Determine the list of volume elements interfering with each cell of the environment.

- Point-Volume Classification : Given any point in the environment determine whether the point is inside/outside the volume element.
- Extinction Coefficient Computation : Given any point inside a volume the extinction coefficient must be known or must be easily computed.
- Volume Sampling : Given an emitting volume choose sample points within the volume in accordance with the emission strength distribution.

Volume modeling is currently a very active area of research[22] and any of the volume modeling techniques described could be used provided the model data enables us to efficiently carry out the computations listed above. For the express purpose of testing out the above algorithm we have implemented the following:

All Pervading Volume: This models a homogeneous absorbing/scattering medium occupying the whole environment of interest. All the solid objects bounded by their surfaces are placed within this medium. This volume interferes with every cell and every point of interest in the environment lies within this volume and has the same extinction coefficient.

Volume bounded by Quadric Surfaces: These model a homogeneous medium enclosed within quadric surfaces. Each volume is specified by its canonical quadric and a 3D transformation. To classify a point with respect to the volume we first apply the inverse of the transformation associated with the volume and then substitute the coordinates of the point in the implicit algebraic form of the associated canonical quadric equation. Interference with cells is also similarly determined and is quite straight forward. Extension to a non-homogeneous medium is also possible if the extinction coefficient is given as a function of the geometric parameters defining the quadric, for example, centre and radius for a sphere.

Data Set: This models a unit cubical volume in a discretised form. The optical properties within the volume are defined by a 3D array $(m \times n \times p)$ with each element of the array holding the value of extinction coefficient, scattering albedo and emission strength if the object is an emitter. Each array element represents a homogeneous medium enclosed within a rectangular box whose dimensions are $(\frac{1}{m}\ \frac{1}{n}\ \frac{1}{p})$. The cubical volume is suitably scaled to the desired size and then is positioned in the environment by applying the appropriate transformations. Classification of a point is carried out once again by applying the inverse transformation and then checking whether the point lies inside the unit cubical extent. Bounds of the object are found by transforming its unit cubical extent. The data set may have been created from physically based simulation results or experimental results or from actual measurements. For an emitting dataset the particle position can be sampled first by discrete sampling the emitter strength distribution among the dataset elements and then for the exact position by carrying out uniform random sampling in the rectangular extent of the element.

For the environment of Fig.1b the participating medium has been modelled as an all pervading medium. Fig.2 shows the images of a plant modelled as a collection of about 161 participating spherical volumes. Fig.3 shows a gaseous emitting volume modeled as a 3D array of rectangular blocks each with its own extinction coefficient, scattering albedo and emission strength.

6 Rendering

There are two important points that must be noted while rendering a scene which includes participating volumes:

- Radiance along the view direction is based on the combined contributions of the particles coming out both from the surface as a result of reflection and from the volume as a result of scattering and/or emission.
- Light passing through a participating volume is attenuated and the expression for the radiance reaching the view point due to the radiance emitted from a distance S away from the view point is given by Bouguer's Equation stated earlier in Section 2.

From the above the radiance reaching the eyepoint can be written as follows:

$$i \;=\; i_v \;+\; i_s$$

In this equation i_v, the cumulative attenuated volume contribution, is the radiance due to each voxel along the path and is given by

$$i_v \;=\; \int_0^{S_{far}} di_{volume} e^{-\int_0^S K(u)du}$$

where S_{far} is the distance along the ray from the eyepoint to the nearest surface or up to the farthest bound of the scene, whichever is shortest, S and u are dummy variables of integration respectively in the range 0 to S_{far} and and in the range 0 to S.
i_s, the attenuated surface contribution, is the radiance due to the nearest surface along the view direction and is given by

$$i_s \;=\; i_{nearest\_surface} e^{-\int_0^{S_{far}} K(u)du}$$

In the absence of any surface along the view direction $i_{nearest\_surface}$ is set to zero. If we make a further assumption that the volume emits uniformly in all the directions[1] then the radiance coming out of the differential volume as given in [20] is:

$$di_{volume} \;=\; \frac{dE_{volume}}{4\pi dA_p}$$

where dA_p is the projected differential volume along the direction of interest. Each *voxel* has been assumed to have uniform interaction property (and hence constant K_{voxel}) and the simulation results have been captured over the whole *voxel*. So if E_{voxel} is the outgoing light energy from the *voxel* then the energy coming out from the unit volume inside the *voxel* is $\frac{E_{voxel}}{V_{voxel}}$ where V_{voxel} is the *voxel* volume. For any cubical differential volume inside the *voxel* of side dS with its two faces normal to the view direction the expression for di_{volume} can now be written in terms of dS as

$$di_{volume} \;=\; \frac{E_{voxel}}{V_{voxel}} dS^3 \frac{1}{4\pi dS^2} \;=\; \frac{E_{voxel}}{4\pi V_{voxel}} dS$$

Substituting the value of di_{volume} in the equation for i_v we get

$$i_v \;=\; \int_0^{S_{far}} e^{-\int_0^S K(u)du} \frac{E_{voxel}}{4\pi V_{voxel}} dS$$

[1] As the illumination from the nonemitting volume is only due to the scattering of light inside the volume this assumption amounts to saying that the volumes are isotropic scatterers.

Coupled with the assumption that the extinction coefficient is constant within a *voxel* we get

$$\begin{aligned} i_v &= \sum_{i=1}^{N} \frac{E_i}{4\pi V_i} \int_{S_{i-1}}^{S_i} e^{-\int_0^S K(u)du} dS \\ &= \sum_{i=1}^{N} e^{-\int_0^{S_{i-1}} K(u)du} \frac{E_i}{4\pi V_i} \int_{S_{i-1}}^{S_i} e^{-K_i \int_{S_{i-1}}^{S} du} dS \\ &= \sum_{i=1}^{N} e^{-\int_0^{S_{i-1}} K(u)du} \frac{E_i}{4\pi V_i} \frac{1 - e^{-K_i \Delta S_i}}{K_i} \\ &= \sum_{i=1}^{N} e^{-\sum_{j=1}^{i-1} K_j \Delta S_j} \frac{E_i}{4\pi V_i} \frac{1 - e^{-K_i \Delta S_i}}{K_i} \end{aligned}$$

Where N is number of *voxel*s along the view direction up to S_{far}, S_i is the distance from the view point to the farthest point of the *i_th voxel* along the view direction with S_0 equal to 0 and ΔS_i is the distance traversed along the view direction inside the *i_th voxel* with ΔS_0 equal to 0.
Similarly we can simplify the light contribution from the nearest surface to get

$$i_s = i_{nearest\_surface} e^{-\sum_{i=1}^{N} K_i \Delta S_i}$$

The algorithm for rendering can now be described as follows:

for each pixel do steps 1 to 6 given below:

1. $sumopacity_\lambda = 0$; $Radiance_\lambda = 0$
2. Define a ray from the eye point through the centre of the pixel.
3. Find the nearest surface along the ray and get $i_{nearest\_surface}$.
4. Get the list of *voxel*s along the ray.
5. While *voxel* list not empty do steps (a) to (d) given below:
   (a) get the next *voxel*.
   (b) $opacity_\lambda = K_\lambda \Delta S_{voxel}$
   (c) $Radiance_\lambda = Radiance_\lambda + \frac{E_{voxel,\lambda}}{4\pi V_{voxel}} \frac{1 - e^{-opacity_\lambda}}{K_\lambda} e^{-sumopacity_\lambda}$
   (d) $sumopacity_\lambda = sumopacity_\lambda + opacity_\lambda$
6. $Radiance_\lambda = Radiance_\lambda + i_{nearest\_surface} e^{-sumopacity_\lambda}$

7 Uncertainty Issues

Random numbers play a major role in any Monte Carlo simulation. However their very random nature introduces uncertainty in the derived results. Hence during the various steps of a Monte Carlo simulation, it is always advisable to reduce the use of random numbers by making direct use of the underlying analytical functions. In this section we shall analyse the different steps of our simulation and wherever possible replace the sampling steps by their analytic equivalents.

In the simulation strategy discussed so far, each sample particle carries a quantum amount of light energy, and contributes an integral multiple (zero or more) of this energy

to the brightness of all the elements of the environment. In fact to most of the elements a sample particle contributes zero and to a very few it contributes a nonzero multiple of its energy. The methods discussed below increase the number of nonzero contributions made by the sample particle to the elements of the environment by allowing a fraction of its energy to be contributed towards their brightness. In these methods the sample particle is no longer assumed to carry a quantum of energy. Rather a sample particle is assumed simply to be a particle carrying a large multiple of energy quanta at a particular wavelength.

7.1 Forced Interaction in Volume

The interaction of the light inside a participating medium is governed by the Bouguer's equation. This equation gives the factor by which the radiance changes after traveling a distance S inside a participating volume. In section 2 we used Bouguer's equation to derive the *pdf* of volume interaction pathlength and sampled that *pdf* to determine whether the sample particle interacts inside the volume element it is passing through. This sampling assures that for a large number of particles entering a volume, the ratio of the particles exiting the volume without undergoing interaction, to the number of particles entering will be equal to the above mentioned factor. The size of this number of particles is highly dependent on the extinction coefficient of the interacting volume and on the maximum distance the particle can travel inside the volume. If the number of particles entering the volume are not large enough then there can be very wide deviation from the expected number of particles interacting inside that particular volume. Since a part of these interacting particles are going to contribute towards the brightness of the volume elements, this sampling procedure is likely to introduce errors in the final illumination computation.

The variance reduction method of *Forced Interaction*[1] avoids this sampling problem by forcing the sample particle to interact with each of the volume elements it is passing through, in the process losing a part of its energy to the volume and exiting the volume with its energy reduced exactly by the amount lost inside the volume. To satisfy the requirement of the Bouguer's equation if W is the energy associated with the the particle entering the volume then the energy of the particle leaving the volume is $We^{-opacity}$ and the energy lost in the volume is $W(1 - e^{-opacity})$. This energy loss is either due to absorption or scattering. Because there is a further decision of absorption or scattering of this energy and if it scatters then that of the direction, we shall pretend as if another particle, carrying energy $W(1 - e^{-opacity})$, is generated inside the volume and subjected to further sampling. The position where this new particle is generated is derived as follows:

> We know that the interaction function in a participating medium is exponential in nature, that this particle must interact inside the volume and that the extinction coefficient, K, is constant inside the volume. If S is the length of the particle track inside the volume then we have the following conditions.
>
> $$pdf \;=\; Ce^{-Ku}$$
>
> where C is some constant, and
>
> $$\int_0^S pdf\ du \;=\; 1$$
>
> Solving for C from the above two equations we get $C \;=\; \frac{K}{1-e^{-KS}}$ and hence
>
> $$pdf \;=\; \frac{Ke^{-Ku}}{1 - e^{-KS}}$$

Thus

$$cdf \;=\; \int_0^x \frac{Ke^{-Ku}}{1-e^{-KS}}du \;=\; \xi$$

Solving for x, the distance of interaction inside the volume element, we get

$$x \;=\; -\frac{1}{K}ln\left(1-\xi(1-e^{-KS})\right)$$

The introduction of this modification results in the following overheads:
a) The number of particles is increased, as for every particle entering the volume two particles result, one exiting the volume and the other interacting with the volume. This increase if unconstrained may result in very rapid particle growth.
b) For every particle entering a volume element additional computation is required for random number generation, evaluation of a logarithmic function and an exponential function. Thus this method must ideally be used selectively for those volume elements where the number of particles entering the volume is known to be small. Fig.4 shows the visual improvement in the simulation result by using the forced collision method. In both illustrations, for forced collision method the number of primary particles chosen has been such that the total simulation time using either of the methods remains the same.

7.2 Absorption Suppression

Light interacting with the elements of the environment gets absorbed and reflected from an opaque surface element or scattered from a volume element. The fraction of the light that is not absorbed is determined by the reflection coefficient in case of the interaction with the surface and by the scattering albedo in case of the interaction with the volume. We have used these properties to define a discrete *pdf* of two events and carried out its sampling to decide on the type of interaction for each interacting particle during the course of the simulation. Again like any other sampling process if the number of particles interacting with the surface or volume element is not large enough then the distribution of the absorbed particles and of the surviving particles will not match the sampled discrete *pdf*. This process can introduce errors into the illumination results of the individual surface or volume element.

The *Absorption Suppression*[1] method, avoids this error by assuming that unless the reflection coefficient (or the scattering albedo) is zero, a particle interacting with the surface (or volume) is always reflected (or scattered) with its energy content reduced to a value equal to the original energy content times the reflection coefficient (or scattering albedo). But by its very definition, in this method the particle tracking will never terminate even for a single particle unless there are completely absorbing elements in the environment or unless the particle is allowed to escape at the system boundaries. However, one can use unbiased terminating techniques like Russian Roulette to remove particles whose weight fall below certain small threshold.

In the algorithm presented in Section 2, a particle interacting with a surface or volume is randomly absorbed if a uniform random number drawn at that point is greater than k_{refl} in case of surface or scattering albedo(Ω) in case of volume. This is based on the assumption that the interaction is a discrete distribution of two events: for surface reflection and absorption with distributions $(k_{refl}, 1-k_{refl})$ respectively and for volume scattering and absorption with distribution $(\Omega, 1-\Omega)$. Every time a particle is reflected it contributes the equivalent of 100% of its energy to the receiver surface/volume brightness and then continues its flight in the reflected/scattered direction. Given this model of interaction, if we consider

an enclosure with all surfaces having uniform reflectance k_{refl} and no participating medium, the probability of a particle undergoing the first reflection is k_{refl}, the second reflection k^2_{refl}, the third reflection k^3_{refl} and so on. So the average relative brightness contribution made by each particle to the given enclosure is given ideally by the factor:

$$B_{av} \;=\; 1 + k_{refl} + k^2_{refl} + k^3_{refl} + \ldots = \frac{1}{1 - k_{refl}}$$

If this value of B_{av} has to result from simulation then in principle one needs an infinite number of samples. Simulation using a finite number of samples will have high variance in the computed equilibrium illumination of the environment.

In column II of Table 2 we have summarised the average brightness contribution made by a particle in a test environment of a cube with all surfaces having a surface reflectance of 0.9. It must be noted that this contribution factor should ideally be 10. As can be seen from Table 2 the variation from this ideal value reduces with increasing number of samples and only by about a million samples is the ideal value almost reached.

Using the absorption supression method a slight change is required in the particle flux capture and the interpretation processes. Instead of keeping a count of the outgoing particles, a cumulative value of weight is maintained for each outgoing particle leaving the surface/volume and on each interaction the particle's weight is scaled down by a factor equal to the reflectance of the surface if the interaction is on surface and by a factor equal to the scattering albedo if the interaction is in volume. Thus flux computation changes to:

$$Outgoing\ Flux = \frac{Total\ weight\ of\ the\ Particles\ Leaving\ the\ Patch}{Total\ Number\ of\ Particles\ Emitted} * Total\ Source\ Strength$$

The result of this technique for the test enclosure with 0.001 as threshold for cutoff and 50% reduction of the particle population below this threshold by Russian Roulette, has been shown in column III of Table 2. The results show a major improvement in the average particle contribution, specially when compared to the results computed with smaller number of samples. However, it is intuitively clear that this method of absorption suppression is computation intensive as each particle is always carried through its reflection history till its weight reaches the threshold and is carried further if it survives the Russian Roulette. It is therefore more appropriate to compare the result produced with equal computational efforts (Column IV of Table 2). The computational efforts have been measured by the number of ray-surface intersections. These results too show an improvement over the simple absorption method.

We have incorporated both the methods, the simple particle absorption and the absorption supression, in our implementation and have often observed marginal visual improvements in the images created using the latter method for equivalent computational efforts. However the improvements have been found to be significant for indirectly illuminated surfaces. We also feel that as the complexity of the 3D configuration increases this variance reduction technique will yield even better results.

7.3 Particle Divergence

There is another sampling step in our simulation which can introduce errors into the simulation results because of the problem of insufficient sampling. This step is the sampling for the outgoing direction for every reflecting or scattering particle. In the absence of enough outgoing particles from the surface or volume, the choice of a single direction for each

reflected or scattered sample particle may result in a very poor representation of all the directions seen by the surface or volume. One possible solution to this problem is what we shall term as the *Particle Divergence Method* in which we sample a number of directions for every single outgoing particle.

In the particle divergence method, an outgoing particle is split into many sub-particles. For each such sub-particle a direction is chosen by sampling the directional distribution of the reflection (or scattering) and the sub-particle is assigned a fractional amount of energy of the original outgoing particle such that the total energy content of the sub-particles is equal to that of the parent particle. This means, if a particle with energy W is split into n sub-particles then each sub-particle is assigned energy $\frac{W}{n}$. Each of the sub-particles is then independently tracked to follow its history.

However, this method causes severe particle multiplication effect, and unless used judiciously will be excessively time consuming. The techniques to reduce this particle population are to use Russian Roulette to selectively terminate the particles with smaller energy content and to make the number of sub-particles generated as being proportional to the energy of the parent particle.

8 Conclusion

The global illumination computed using Monte Carlo based methods will fluctuate around the *real* value. The fluctuation can be reduced by increasing the number of particles used in the simulation. In general it is not possible to ascribe a 100% confidence to the values that we obtain due to the statistical uncertainty inherent in the methods. The analytical methods also are not free of this problem. Numerical uncertainties arise not only due to the discretisation of the shapes and directions but also from the fact that various simplifying assumptions regarding the environment have to be made to make the analytical solutions computationally feasible. There are no known methods to estimate such numerical uncertainties. In particular when dealing with general environments and participating volumes optimal discretisation is absolutely essential. Otherwise both accuracy and efficiency are affected. Monte Carlo simulation does not depend on the discretisation and can deal with more complex environments without undue increase in computational expenses. With computing costs continuously decreasing the Monte Carlo simulation technique is a viable alternative to other deterministic approaches. And for complex environments and participating volumes it may well be the only one.

9 Acknowledgements

We are grateful to Dr S Ramani, Director, NCST for his support and encouragement.

References

[1] E. E. Lewis and W.F. Miller Jr. *Computational Methods of Neutron Transport.* John Wiley & Sons, 1984.

[2] J. R. Howell. Application of monte carlo to heat transfer problems. *Advances in Heat Transfer*, 5:1–54, 1968.

[3] S. N. Pattanaik and S. P. Mudur. Computation of global illumination by monte carlo simulation of the particle model of light. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 71–83, Bristol, UK, May 1992.

[4] James Blinn. Light reflection functions for simulation of clouds & dusty surfaces. *Computer Graphics (SIGGRAPH '82 Proceedings)*, 16(3):21–26, 1982.

[5] James T. Kajiya and Brian P. Von Herzen. Ray tracing volume densities. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):165–174, July 1984.

[6] Nelson L. Max. Atmospheric illumination and shadows. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):117–124, Aug. 1986.

[7] Tomoyuki Nishita, Yashuhiro Miyawaki, and Eihachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21(4):303–310, July 1987.

[8] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21(4):293–302, July 1987.

[9] Michael Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22(4):75–84, Aug. 1988.

[10] Subdeb Chattopadhyay and Akira Fujimoto. Bi-directional ray tracing. In *Computer Graphics 1987 (Proceedings of CG International '87)*, pages 335–343. 1987.

[11] Andrew Glassner. *An Introduction to Ray Tracing*. Academic Press, 1989.

[12] A.T. Campbell, III, and Donald S. Fussell. Adaptive mesh generation for global diffuse illumination. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4):155–164, Aug. 1990.

[13] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):51–60, July 1991.

[14] S.P. Mudur and S.N. Pattanaik. Multidimensional illumination functions for visualisation of complex 3d environments. *The Journal of Visualisation and Computer Animation*, 1(2):49–58, 1990.

[15] Channing P. Verbeck and Donald Greenberg. A comprehensive light-source description for computer graphics. *IEEE Computer Graphics and Applications*, 4(3):213–222, 1984.

[16] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):212–222, July 1984.

[17] Peter Shirley. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, 1990.

[18] Gregory J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(4):265–272, 1992.

[19] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods. Volume I.* John Wiley & Sons, 1986.

[20] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer.* Hemisphere Publishing Corporation, 1981.

[21] A. Fujimoto, T. Tanaka, and K. Iwata. Arts: Accelerated ray-tracing system. *IEEE Computer Graphics and Applications*, 6(4):16–26, 1986.

[22] Arie Kaufman. 3d volume visualisation. In G. Garcia and I. Herman, editors, *Advances in Computer Graphics VI. Image: Synthesis, Analysis & Interaction*, pages 175–203. Springer Verlag, 1991.

Without Participating Medium With Participating Medium

Figure 1: A 3D environment with spherical light sources.

Figure 2: A plant modeled as a participating volume with around 161 spherical volume elements.

Figure 3: A Gaseous Emitting Volume.

Without Forced Interaction. With Forced Interaction.

Figure 4: A 3D enclosure with participating medium illuminated by a directional light source.

Geometry	pdf_v	Equation for Sampling v	Equation for Computing Position
Triangle	$2(1-v)$	$1-\sqrt{\xi_v}$	$P_0+(1-v)u(P_1-P_0)+v(P_2-P_0)$
Rectangle	1	ξ_v	$(1-u-v)P_{00}+uP_{10}+vP_{01}$
Sphere	$\frac{1}{2}\pi sin\pi v$	$\frac{cos^{-1}(1-2\xi_v)}{\pi}$	$(Rsin\pi v.cos2\pi u, Rsin\pi v.sin2\pi u, Rcos\pi v)$, for a sphere with centre at origin.
Cylinder	1	ξ_v	$(R.cos2\pi u, R.sin2\pi u, v.H)$, for a cylinder with one end at origin and axis along the +Z-direction
Cone	$\frac{2(R_0+(R_1-R_0)v)}{R_0+R_1}$	$\frac{\left(-R_0+\sqrt{R_0^2+\xi_v(R_1^2-R_0^2)}\right)}{R_1-R_0}$	$(r.cos2\pi u, r.sin2\pi u, v.H)$ where $r=R_0+v(R_1-R_0)$, for a cone with one end at origin and axis along +Z direction.
Disc	$\frac{2(R_0+(R_1-R_0)v)}{R_0+R_1}$	$\frac{\left(-R_0+\sqrt{R_0^2+\xi_v(R_1^2-R_0^2)}\right)}{R_1-R_0}$	$(r.cos2\pi u, r.sin2\pi u, 0)$ where $r=R_0+v(R_1-R_0)$,for a disc with center at origin and normal along +Z direction.

Table 1: Position sampling equations.

Particles	*Simple Absorption(SA)*	*Absorption Supression(AS)*	*AS with Equiv Work of SA*
100	10.35	9.998	9.996
1000	9.852	10.00	9.998
10000	10.06	10.00	10.00
100000	9.957	10.00	10.00
1000000	9.991	10.00	10.00

Table 2: Relative Performances of simulations based on Simple Absorption and Absorption Supression.