Computation of Global Illumination by Monte Carlo Simulation of the Particle Model of Light

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

Abstract

In the particle model, light is assumed to be composed of photons originating at the light sources and moving within the environment, possibly getting scattered by the medium and refracted or reflected by the objects in the environment until they are eventually absorbed. Using Monte Carlo methods for simulating the behaviour and keeping track of a sufficiently large sample of photons in the environment, illumination effects can be estimated.

In this paper we first describe the algorithm used for Monte Carlo simulation and show how the results are used for computing global illumination effects and for synthesising realistic images. A distinctive feature of the algorithm is that as the simulation progresses, with increasing number of samples, the computed illumination is progressively refined. A detailed comparative performance analysis with a straightforward implementation of the radiosity method shows that the results are equally good and the computation times comparable. The major advantages over the other well established methods of *ray tracing* and *radiosity* stem from the fact that complex environmental behaviour can be more easily accommodated. In particular we discuss techniques for dealing with complex extended light sources and large unevenly lighted surfaces without any need for apriori discretisation, and the issues related to the rendering of non diffuse environments. Finally we describe a variation in the simulation algorithm for variance reduction and compare the results with the earlier algorithm.

1 Introduction

Two different models - the *Wave Model* and the *Particle Model* - are in use for explaining the behaviour of light. In the wave model light is assumed to be electromagnetic waves resulting from the oscillation of charged particles, whereas in the particle model light is assumed to be composed of small packets of energy moving at a speed of light. For some aspects of the behaviour of light these models are complementary to each other. For example, the propagation of light is explained better by the wave model, whereas the interaction of light with other matter is better explained using the particle model. In the study of light either of the models may be used depending on whichever one is more suitable for dealing with the behavioural aspect that is of interest.

The illumination at any point in a complex three dimensional environment is determined not merely by the light received directly from the light sources but also by the light received indirectly due to scattering, refraction and reflection, that is, its interaction with the medium and the objects in the environment. This illumination is aptly termed global illumination. That the accurate computation of global illumination leads to synthesis of more realistic images is by now well recognised in the field of computer graphics [8]. Throughout this paper we shall be concerned only with the particle model of light and its use for determining the global illumination in a complex three dimensional environment.

In the particle model, the packets of energy, referred to as *photon*s are shot out or emitted in different directions from different positions on the surfaces of the light sources. Photons move in a straight path and may come across obstacles, hence forward termed receivers. At the surface of the receiver a photon may undergo absorption, thereby loosing all its energy to the receiver or may undergo reflection thereby changing its direction. The new direction acquired by the photon is determined by the direction from which the photon hits the receiver and the bidirectional reflection function of the receiver surface. The photon continues its flight until it is absorbed. If we assume that there is no change in the emissive behaviour of a light source over a period of time then because of the above mentioned process an equilibrium is established in the rate at which photons leave out surfaces in an environment. This rate (flux) determines the brightness of the objects. The energy of a photon is directly related to its wavelength, a term by which we often characterise a photon. Each wavelength in the range of 400nm to 770nm is associated with a unique colour. The distribution of the wavelengths among the outgoing photons from the surface of an object determine the colour of the object. Using this model, global illumination is determined by finding the photon flux per unit area at different wavelengths for each of the surfaces of the objects in the environment.

In complex environments where the analytical equations get extremely complex to solve, this photon flux can be estimated by carrying out a time-independent simulation of the behaviour of a sufficiently large sample of photons and keeping track of their histories. As has been shown in other disciplines such as the Neutron transport [12] or Heat Transfer [9], such simulations are best carried out using Monte Carlo methods. Although in Computer Graphics, Monte Carlo methods have been applied to stochastic ray tracing and to find solutions to the rendering equation [10, 4, 1] it is only in [15] that there is some reference to its use for computing illumination but the primary emphasis is on time complexity analysis. To the best of our knowledge ours is the first in depth study of the use of the Monte Carlo simulation for computing global illumination.

A Monte Carlo simulation is carried out using a finite number of samples, in this case photons. During the process of simulation each photon is associated with a light source, i.e. an emitter surface and is assigned a position on the emitter surface, a wavelength and a direction. All assignments are done making random choices which over a sufficiently large sample would match the actual behaviour. Such assignments in Monte Carlo simulation are carried out by random sampling of the probability distribution functions (*pdf*) representing the behaviours [11]. Thus for example, the probability of a photon being associated with a more powerful light source is higher than its being associated with a light source of lesser power. Similarly the other assignments above, would all depend on the shape and emission properties of the emitter surface. For each emitted photon, the nearest receiver in the environment along its associated direction is then determined. At the receiver surface the photon is absorbed or reflected. The probability of either of these happening is determined by the surface reflectance. If the photon gets reflected then the photon is registered in the outgoing flux of the receiver surface. The photon is then assigned a new direction and once again the nearest receiver surface along its new path is determined. This process is termed *photon tracking*. For every photon generated this photon tracking is continued till it is absorbed.

On completion of tracking of all the photons we get a simulated photon flux on all the surfaces in the environment. This simulated flux is an estimate of the actual flux in the real environment. From principles of statistics we know that the larger the number of samples the better is the agreement of the estimator with the actual value. Since it is not clear how large is large we have carried out extensive experiments with our implementation to get a feel for the number of samples required. We have compared the results as a function of number of samples with the results obtained from an analytical method and found that results match very well. In the rest of this paper we shall discuss in detail our implementation of the Monte Carlo simulation method for computing global illumination. Not only do we find that it matches in performance with other well established methods but more importantly we find that the method can be easily extended to deal with geometrically and optically complex environments without undue increase in computation time. We describe how complex light sources and large unevenly lit surfaces are handled. We also discuss some issues related to rendering in the synthesis of images from the simulation results and end with a detailed analysis of a variation in the basic simulation algorithm for variance reduction used by many Monte Carlo programs[12, 9].

2 The Monte Carlo Simulation Method

2.1 The Algorithm

Throughout this paper we consider three dimensional environments consisting of light sources and other objects with the following assumptions:

- The medium in the environment does not interact with the photons.
- The objects are all opaque and are described by their bounding surfaces and associated optical properties such as emissivity and bidirectional reflectance.

For such environments we have used the algorithm described below to carry out the Monte Carlo simulation:

Step–I Decide on the total number of Photons to be used in the Simulation.

Step–II From the environment specification compute the total strength of the emitters in the environment at each wavelength.

Step–III For each photon repeat steps (1) to (6) below:

1. Choose the photon's wavelength by *Discrete Distribution Sampling* of the cumulative emission spectrum.
   In the presence of multiple light sources choose the emitter from which the photon will originate by *Discrete Distribution Sampling* the emitter strength distribution at the chosen wavelength.
2. Choose the position on the emitter at which the photon originates by position sampling the emitter surface geometry.
3. Increment the *Emission Count* of the emitter.
4. Update the outgoing photon flux at the emitter surface.
5. Choose the direction in which the photon is emitted by sampling the emitter surface emissivity.
6. Repeat steps (a) to (c) below until the photon is absorbed.
   (a) Find the nearest surface along the photon path.
   (b) Choose the type of interaction i.e. absorption or reflection by *discrete sampling* of surface interaction distribution.
   (c) If the interaction is Reflection
      i. Update the outgoing photon flux on the reflecting surface.
      ii. Assign the reflection direction by sampling the *bdrf* of the reflecting surface.

Step–IV Interpret simulation results.

In the algorithm above we have referred to a number of sampling principles. They are briefly discussed in the Appendix. Section 3 discusses in detail position sampling for various emitter surface types (step III.2) and direction sampling for emitted photons based on the emissivity of the source (step III.5). Since the path of a photon is assumed straight, finding the nearest surface hit by the photon (step III.6.a) is done by carrying out ray-surface intersections which has been researched extensively in computer graphics [6]. Section 4 discusses the method of updating the outgoing flux (step III.4, III.6.c.i) on large and complex surfaces. Sampling of *bdrf* (step III.6.b) is also discussed in the same section.

2.1.1 Computing Flux Density

The interpretation of the simulation result is carried out as follows:
From the outgoing flux recorded at each surface we can obtain for any subpatch of the surface a count of outgoing photons due to each emitter at each wavelength. This count is the estimator of the equilibrium photon flux density of that subpatch. The relation is as follows:

$$Outgoing\ Flux\ due\ to\ each\ light\ source =$$

$$\frac{Number\ of\ Photons\ due\ to\ a\ Source}{Number\ of\ Photons\ Emitted\ from\ the\ Source} * Source\ Strength$$

$$Total\ outgoing\ Flux \ = \ \sum Outgoing\ Flux\ due\ to\ each\ light\ source$$

where the flux has the unit of *energy per unit time.*

$$Flux\ Density \ = \ \frac{Total\ outgoing\ Flux}{Patch\ Area}$$

where the flux density has the unit of *energy per unit area per unit time.*

2.2 Progressive Refinement

The complexity of the environment as well as the nature of the values that are to be estimated would both determine the number of samples that are needed to make reasonably accurate estimates. Hence deciding on an optimal number of samples for carrying out the simulation is not only difficult but also highly dependent on what interpretations of the simulation results are needed. However, a distinguishing feature in the design of the above algorithm is that it is truly progressive in nature. At any instant of time, after a couple of thousand photons or so have been tracked, all the light sources and the other bright surfaces in the environment would have made contributions corresponding to their actual behaviour. Thus after a reasonable number of photons have been tracked we have illumination information which is not likely to be drastically different from the result derived after a larger number of photons are tracked. This could be used to great advantage in adaptively deciding on the number of samples to be used in the simulation.

If we visualise the simulation and the interpretation of simulation results as two different processes communicating with each other then the *simulator process* could transfer simulation results to the *interpreter process*, say, in bundles of a thousand photons or so and the interpreter process could in turn check if there is no appreciable change over a period of time and signal back to the simulator process to terminate the simulation. Similarly, we can create images at intermediate stages of the simulation and each image will be more refined than its predecessors. If the main purpose of the simulation is image synthesis then the process can be terminated when visually satisfactory results have been arrived at.

2.3 Comparison with Radiosity Method

For comparison purposes, we have carried out a simple implementation of Full Radiosity solution using the hemicube method [2] for form-factor computation and the iterative matrix solution method for carrying out final radiosity computation to produce radiosity values over the surface patches in the environment. The term radiosity by definition is the photon flux per unit area and hence is the same as that computed in the Monte Carlo simulation method. In order to be able to carry out the comparison by tabulating the results we have chosen a simple convex environment, a cubical enclosure with one surface as an emitter emitting at a single wavelength and all surfaces having a uniform reflectance of 0.9 at that wavelength. We have uniformly subdivided each face to a 10 x 10 grid for more accurate form-factor computation for use in the radiosity method. In both the cases we have computed the average radiosity at each of the faces of the cube. Table.1 compares the results obtained from simulation with different number of samples with results from the radiosity methods. Flux density for the faces has been normalised to a maximum of 1 at the emitter surface. In Fig.1(a) we have shown the comparison of a test environment in the form of images rendered by the radiosity method and the Monte Carlo method. The environment contains a total of 113 polygonal surfaces with larger surfaces selectively broken down to a 10 x 10 grid of patches. Fig.1(b) shows the images rendered using the simulation results after the tracking of 1000, 10,000 and 1,000,000 photons respectively.

As can be seen from these images the Monte Carlo simulation method produces results which compare very well with those of the radiosity method. The computation times [1] also are of the same order (282 seconds for radiosity vs 309 seconds for the 100,000 sample Monte Carlo simulation). In fact we expect that as the number of surfaces in the environment increases, the

[1] These times would certainly improve if hardware acceleration techniques were to be used for scan conversion and ray tracing.

performance of the Monte Carlo simulation method will be generally superior. The more significant advantages stem from the fact that the Monte Carlo method is inherently capable of handling far more complex three dimensional configurations (both geometrical and optical complexity) with greater flexibility, simplicity and speed. In this paper we discuss in detail the simulation for environments containing complex extended light sources and also large and complex reflecting surfaces. In a companion paper we are presenting our extensions of this method for dealing with environments with a participating medium [14].

3 Complex Light Sources

Light sources, i.e., emitters play a crucial role in the illumination of the environment. There are mainly three important characteristics of the emitter that influence the illumination [17]. They are *geometry* (emitter surface shape), *spectral distribution* (emitter strength at different wavelengths) and *luminous intensity distribution* (emitter strength in different directions). Light source geometries are varied and determine the distribution of light in the environment and also control the nature of the shadows generated. Spectral distribution controls the colour of the objects in the environment. Luminous intensity distribution influences the relative brightness of the objects positioned around the emitters. So far only spectral distribution is handled satisfactorily by the existing rendering methods. Though it is common practice to carry out the illumination computation only with red, green and blue wavelength strengths of the emitter, in principle, one can carry out the computation for more number of wavelengths [8]. The effects of extended light source geometry and anisotropic luminous intensity distribution cannot in general be determined by most of the existing methods. On the other hand Monte Carlo simulation is inherently capable of including the above effects. We explore some of these below.

3.1 Light Source Geometry

Traditionally in computer graphics the visible shape of the light source and its emissive geometry have been treated differently. For emissive geometry highly simplified assumptions are made. 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 [7]. The radiosity method is based on computing geometric form factors between any two surfaces, which is the fraction of the light energy reaching one surface from the other. As discussed in Section.4 in the radiosity method any large or complex surface must be first broken down into small planar patches before its use in illumination computation. Considering that the worst case performance of the radiosity method is $O(N^3)$ for computation time and $O(N^2)$ for storage, where N is the number of surfaces in the environment, this requirement of discretisation into small planar surfaces imposes tremendous computational burdens if complex light source geometries have to be dealt with accurately.

On the other hand in the Monte Carlo simulation method there is not much difficulty in dealing with complex emitter surface shapes directly. In our implementation we have successfully incorporated the following emitter surface geometries: triangles, rectangles/parallelograms, spheres, cylinders, cones and discs. The strategy used is described below in detail. We also believe that more complex shapes such as surfaces of revolution or doubly curved surfaces can be incorporated if the need arises.

As stated earlier during the simulation process all the photons are assigned positions on the emitter surface. If we assume that the emission strength is uniform over the surface of the emitter, then the only requirement for using any arbitrary geometry is that we must be able to generate photons uniformly over the surfaces. In other words the photon density (photons/unit area) must be same throughout the emitter's surface. So the essence lies in devising the proper sampling strategy. In the following discussion each point on a surface is uniquely represented by two parameters (u and v) which are independent of the position of the light source.

Let us consider first the rectangle or parallelogram and cylinder shaped surfaces as they are the simplest ones to sample. Consider a rectangle or parallelogram with its four corner points defined by P_{00}, P_{10}, P_{11}, and P_{01} all positions in three dimension. Every point on the bounded rectangular surface is given by the vector equation:

$$P = (1 - u - v)P_{00} + uP_{10} + vP_{01} \;\ldots.\; (0 \leq u, v \leq 1)$$

It is easy to see that by choosing uniform random numbers in the range 0 to 1 for u and for v automatically results in uniform photon distribution over the surface of the emitter. The exact position in 3D space is computed by substituting the values for u and v in the above equation.
Similarly a cylinder with base radius R and height H can be described as a

biparametric surface with each point represented by (u,v) where u is related to the circumferential angle ϕ, by u=$\phi/2\pi$ and v is related to the height, h, at that point by v = h/H. As the surface is symmetric along the circumferential direction and uniform along the height, uniform random numbers in the range of 0 to 1 for u and for v will give uniform density of photons. For a cylinder positioned at the origin with its axis along the positive Z-direction, the exact point on the surface is given by substituting the values of u and v in the equation below:

$$P =< R.cos2\pi u, R.sin2\pi u, v.H >$$

The generic method used to derive the sampling equations from biparametric representations of the triangle, sphere, cone and disc is as follows: (The u and v parameter directions for each of these are shown in Fig.2)

i) Sample along the u parameter by assigning a uniform random number in the range 0 to 1 to the u parameter.
ii) Sample along the v parameter using the principle of *transformation of random variable* (*cf* Appendix). In this method the first task is to formulate the *pdf* of v. For this each of these geometries is assumed to be composed of differential strips. As shown in Fig.2(a), for the triangle the strip is rectangular with width b and height dh, for the sphere and the cone the strip is cylindrical with base radius r and height dL and for the disc the strip is an anular ring with radius r and thickness dr. The probability of a photon coming out from within a strip is:

$$p(v)dv = \frac{Area\ of\ the\ strip}{Area\ of\ the\ Whole\ Surface} \quad and \quad \int_0^1 p(v)dv = 1$$

Thus p(v) is the *pdf* of the v parameter. So from the principle of *transformation of random variable*

$$Uniform\ Random\ Variable(\xi_v) \;=\; cdf \;=\; \int_0^v pdf\ dv$$

The solution to this equation gives the value of v in terms of the uniform random number ξ_v. We give below the derivation for a triangle.

- Area of the Triangle = *1/2 B . H*
- Area of the strip = *b . dh*

- From similar triangle principle : $\frac{b}{B} = 1 - \frac{h}{H}$ *or* $b = B(1 - \frac{h}{H})$
For the triangle the v parameter is along the height and can be defined by the relation $v = \frac{h}{H}$. Substituting h and dh we have the area of the strip $= B(1-v)H.dv$

- So $p(v)dv = 2(1-v)dv$ *and* $cdf = \int_0^v 2(1-v)dv$

- So $v = 1 - \sqrt{1-\xi_v}$. However, ξ_v being a uniform random variable in the range of 0 to 1, $(1-\xi_v)$ is also an uniform random variable in the same range. So we can write $v = 1 - \sqrt{\xi_v}$.

For each of the shapes supported in our implementation Table.2 shows the pdf_v, the sampling equation for v and the equation for computing the exact position on the surface. Extension of this generic method to more analytically complex surfaces is based on the observation that a uniform photon density over the entire surface also means uniform photon density over the smaller parts of the surface. The distribution of photons amongst the various parts constituting a surface will be in proportion to the individual area. So the strategy for these surfaces is to first subdivide into smaller parts of supported shapes, then compute their fractional area, carry out *discrete sampling* of the fractional area distribution among the parts to choose the part and finally carry out uniform position sampling in the selected part to choose the exact position from which the photon will be generated.

3.2 Luminous Intensity Distribution

Generally in computer graphics emitters are restricted to have uniform intensity distribution in the hemispherical direction. Assuming point light sources, extensions have been proposed in the published literature to support emission in a small range of directions or to support emission following a cosine law for attenuation [20]. However, we are not aware of any such extensions to the radiosity method which deals with area light sources. Just as in the case of complex light source geometry, directional intensity distribution can also be supported quite easily in Monte Carlo simulation. Supporting any complex emissivity means assigning a direction to each emitted photon in such a way that the distribution of the photon samples match with the emitter's intensity distribution.

A direction is defined by a pair (ϕ, θ) where ϕ is the circumferential angle and θ is the cone angle. In most cases we make the assumption that the emission does not depend on the circumferential angle ϕ and hence intensity distribution is specified only along the cone angle by goniometric diagrams [17]. 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* (*cf* Appendix) of the Goniometer Curve.

In this paper we consider the direction sampling procedures for two different intensity distributions: *diffuse emission* and *fixed directional emission.* In the latter distribution the emission is only along one direction and hence in such an emitter the generated photon is assigned directly the only predefined direction associated with the emitter surface. Such type of emitters may be used for modeling sun light coming through the window pane.

Direction Sampling for Diffuse Emitters In diffuse emission the emitted intensity is uniform in all directions and total emissive power per unit area is π times the emission intensity. With ϕ sampled as explained in the above paragraph, the sampling procedure for θ is as follows:
Let the hemisphere in the Fig.2(b) represent the hemispherical emission direction around the differential patch, P, with area dA positioned at the center of the hemisphere. The probability $p(\theta)d\theta$ of the light photon reaching the differential cylinder, C, at θ which makes a solid angle dω at the centre of P is the fraction of the light energy that reaches C on being diffusely emitted from the patch.
The light reaching C from P= $2.\pi.i.dA.cos\theta.sin\theta.d\theta$
The total light emitted from $P = \pi.i.dA$. So

$$p(\theta)d\theta = \frac{2.\pi.i.dA.cos\theta.sin\theta.d\theta}{\pi.i.dA} = 2.cos\theta.sin\theta.d\theta$$

By the principle of *Transformation of Random Variables*:

$$\xi_\theta \;=\; \int_0^\theta p(\theta)d\theta \;=\; 2\int_0^\theta cos\theta sin\theta d\theta$$

and solving the equation we get $sin\theta = \sqrt{\xi_\theta}$. Thus for the photon emitted from a diffuse emitter the direction assigned is 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.

4 Illumination Over Large and Complex Receivers

Any 3D environment would include large surfaces like walls, floors, ceilings, table tops etc. as well as complex surfaces like lamp shades, chair backs, flower vases and other such objects. And all these surfaces, large or small, simple or complex, receive and reflect light. In fact light sources themselves may receive light emitted or reflected by other surfaces. In general the reflectance properties of different objects would not be the same and hence a wide variety of reflectance behaviours have to be considered when determining global illumination. It is extremely important to recognise here that in most situations, the surfaces in a 3D environment would not be uniformly illuminated. There could be shadows and there could be continuous change in illumination over the entire surface. Hence a global illumination method must be able to deal with the following:

- A range of surface geometries.
- A variety of reflectance properties.
- Non uniform illumination over surfaces.

Of the two main methods for global illumination computation: ray tracing and radiosity, ray tracing has the distinction of handling a wide variety of geometries and of accurately handling specular reflectance property. Ray tracing has also been extended to handle diffuse reflectance property but at the expense of heavy increases in computation time [10, 19]. The Radiosity method on the other hand can basically handle diffuse reflectance property accurately. It has also been extended to handle pure specular reflectance behaviour. However, there are two very basic assumptions in the Radiosity method.

1. The surface patches have uniform illumination. Otherwise the form-factor equation becomes very complicated and hence cannot easily be computed.

2. The surface patches are small and planar. Otherwise, the solid angle subtended by another surface cannot be assumed to be the same over the entire surface. (This assumption is the basis on which the form-factor for the whole surface is approximated to be the form-factor computed for the differential patch at the center of the surface.)

Therefore in the radiosity method all complex surfaces must be first subdivided and unless proper apriori subdivision of a nonuniformly lit surface is carried out, the illumination gradient over the surface will not be accurately captured (shadow boundaries could be completely missed) [13]. Greenberg et al [3] do suggest a technique of repeatedly determining the illumination and using the results for adaptive discretisation. However the technique does not always guarantee success in capturing the gradient accurately. In the rest of this section we show how in the Monte Carlo simulation method we deal with geometrical and reflection complexity and also discuss how to maintain the flux so that accurate illumination gradient computation is facilitated.

4.1 Complex Analytical Surfaces

The main issue involved in handling analytically complex surfaces in the Monte Carlo method is that it should be possible to find the nearest surface along the photon's path. As the photon travels in a straight line during its flight, finding the nearest receiver, as stated earlier, amounts to computing ray-surface intersection. The ray tracing literature abounds with methods for ray-surface intersection for a large variety of surface shapes [6]. All such surfaces in principle can be therefore used as receivers in the simulation process. In our implementation, we have considered polygons, spheres, cylinders, cones and discs. Other surface geometries can be added without too much effort.

4.2 Complex Surface Reflectance

The reflectance of a receiver determines the distribution of reflected intensity around its surface. In the simulation, surface reflectance is used to choose the reflection direction for a photon. We consider two types of idealised reflectance behaviour - diffuse reflectance and mirror reflectance. In the diffuse reflectance the reflected intensity distribution is uniform around the surface similar to the diffuse emission process discussed in section 3.2. Hence the direction for a reflected photon 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, similar to the direction chosen for a diffusely emitted photon. 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 photon hits the receiver surface the reflection direction is computed from the incident direction and the normal to the surface at that point. Sampling of complex reflectance must be carried out in exactly the same manner as for complex directional emission.

4.3 The Positional Distribution of Photons on a Surface

In the process of simulation a region of the surface under shadow will have very few photons while another region of the same surface directly facing a light source will have a very large number of photons reflected from that region. Thus the positional distribution of the reflected photons directly gives us the variation of brightness over the surface. This is clearly illustrated by the example scene shown in Fig.3(a) along with Fig.3(b) showing the scatter plot of the photons reflected from a surface on the scene. Therefore if in the process of simulation we capture the photon positional distribution then these simulation results can be interpreted not only to determine intensity in a region but also to determine how it varies over the surface.

A naive approach to capture this distribution would be to record the position on the surface, say the parametric values on the surface (u,v) for every reflected photon. The storage requirements in such an approach would however be prohibitive. If the total number of samples is say a million and the average number of reflections undergone by a photon is k, then in total, k million positions would have to be recorded. Interpretation of these simulation results for the purpose of image synthesis would translate to the problem of computing the photon flux for a region of the surface visible through a pixel. If this has to be solved accurately then this is the equivalent of locating all points within a region and once again could make excessive demands on computational resources. Obviously a more compact and efficient method of storing the photon distribution is needed.

In our present implementation the photon distribution is accumulated over a rectangular mesh (in UV space) imposed over each surface. Ideally the mesh structure should be determined automatically and we are continuing experiments in order to determine the strategy to be adopted for defining an optimal mesh structure for each surface in the environment. The choice of the mesh is important for eventually it is this which determines how well the illumination gradient over a surface has been captured for image rendering. This is somewhat analogous to the situation in the radiosity method where the discretisation of the surfaces into constant intensity patches determines the capture of the gradient. However, unlike in radiosity where the discretisation also determines the accuracy of the global illumination computations, 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 and reflectance behaviour without any simplifications or approximations. A further point to be noted is that while in the radiosity method the computation time increases rapidly with increase in the number of patches the Monte Carlo simulation time depends only on the number of photons and the environmental complexity and is independent of the mesh structure.

5 Image Rendering Issues

In computer graphics the primary purpose of computing global illumination is for use in rendering images. To render the image we must be able to compute the intensity of light emitted/reflected by a surface and reaching the eye along the view direction. As discussed earlier, Monte Carlo simulation provides us with flux densities over the surfaces of the scene. The flux density and intensity are related by the general equation:

$$FluxDensity = \int_{\Omega} i(\theta, \phi) cos(\theta) d\omega$$

where $i(\theta, \phi)$ is the intensity along the (θ, ϕ) direction. For a diffuse surface the intensity of reflected light is constant in all directions. With $i(\theta, \phi)$ independent of θ, ϕ the above equation simplifies and we get $i = \frac{FluxDensity}{\pi}$. However, such simplifications are not possible for other types of surfaces. Computing $i(\theta, \phi)$ requires the knowledge of directional distribution of the computed flux density. This would require the capture of outgoing photons from a receiver/emitter surface as a function of direction. Though in principle this may be possible, in practice it will require a very large number of simulation photons for accurately capturing both positional and directional distribution and hence computation and memory overheads will be prohibitive. In order to overcome this problem we have adopted the following strategy.

- The positionally distributed photon flux is maintained in a direction independent manner only over all the diffuse surfaces in the environment.

- Non-diffuse surfaces do participate in full in the simulation. Photons hitting a non-diffuse surface are absorbed or reflected in a manner matching the behaviour of the *bdrf* of the surface. Thus even if the reflected photon flux is not maintained for these surfaces their contributions in the lighting of other surfaces in the environment is properly accounted for.

- At the time of rendering, for visible parts of non-diffuse surfaces, the illumination is computed only in the view direction. This computation is based on the observation that the illumination of a surface which is not an emitter can be determined if the illumination of all the surrounding surfaces is known[10]. Any of the established techniques of scan conversion, subdivision(dicing) or ray tracing could be used for determining visible parts [5]. However, once it is established that through a particular pixel a non diffuse surface is visible then from that surface, intensity in the view direction must be estimated.

- If we assume that non-diffuse surfaces are ideal specular surfaces i.e. mirrors, then the technique used in [18] could be adopted. A ray in the direction mirroring the view direction is shot. If the nearest surface in that direction is a diffuse surface then the illumination is known. Otherwise the process is continued.

- For non-diffuse surfaces with more general reflectance behaviours a more elaborate procedure would be needed for estimating the illumination in the view direction. One possible technique which we are planning to investigate is to carry out a Monte Carlo sampling of the surrounding of the nondiffuse surface and to suitably weight the diffuse intensity incoming from each surrounding direction. The weight to be used would be the value of the bidirectional reflection function for the incoming sample direction and the outgoing view direction. The mirror surface case discussed above is a special case of this method as the value of bidirectional reflection function is non-zero only when the sampled direction is the mirror reflection direction of the view direction. If some of the objects in the sampled directions around the nondiffuse surfaces are also nondiffuse then the intensity from such surfaces has to be computed by applying the sampling method recursively. [2] The strategy works quite well if we assume that the environment is composed of mainly diffuse surfaces with only a few non-diffuse surfaces so that the recursive computations are kept to a minimum.

[2]This strategy of estimating the intensity in the view direction from the surrounding surface illumination would obviously get into trouble if we had two mirrors parallel to each other and the view direction is perpendicular to one of them. However, such a situation would be equally problematic for the ray-tracing and the radiosity techniques as well.

6 A Variation in the Simulation Algorithm

In the algorithm presented in Section 2.1, a photon intersecting a surface is randomly absorbed if a uniform random number drawn at that point is greater than k_{refl}. This is based on the assumption that the interaction is a discrete distribution of two events: reflection and absorption with distributions $(k_{refl}, 1 - k_{refl})$ respectively. Hence the method of absorbing the photon is in accordance to the principle of *Discrete Distribution Sampling*. Every time a photon is reflected it contributes the equivalent of 100% of its energy to the receiver surface brightness and then continues its flight in the reflected direction. If we consider an enclosure with all surfaces having uniform reflectance k_{refl}, the probability of a photon 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 photon 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 may suffer from statistical uncertainty in the computed equilibrium illumination of the environment. This uncertainty is called *high variance*.

In column II of Table.3 we have summarised the average brightness contribution made by a photon in our test environment of section 2.2, 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 evident from Table.3 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.

A variance reduction technique that is used by many Monte Carlo programs in other disciplines [12, 9] is based on the following idea: On every interaction instead of choosing to absorb the photon the photon's brightness contributing capacity (*weight*) is reduced and the photon is always allowed to continue. The photon starts with a weight of 1 from the source. On every encounter with a surface its weight is reduced by a factor of k_{refl} and this modified fractional weight is contributed to the brightness of the reflecting surface. However, direct use of this method is impractical, as the tracking of a single photon never terminates. Termination of this process as the photon's weight falls below a threshold is a practical solution, but will introduce a systematic negative bias into the system. A statistical technique called Russian Roulette [3] may be used to reduce this bias. The technique is as follows:

> As the weight of the photon falls below the threshold Russian Roulette is played to decide whether it should be terminated or not. If the photon survives the termination then the photon is allowed to continue with increased weight.

The computational equivalent of the Russian Roulette [12] is to choose a predefined number $\mathcal{N}$ between 2 to 10. Once the weight of the photon reduces below a sufficiently small threshold a uniform random number ξ is generated. The photon is removed from the system only if $\xi > 1/\mathcal{N}$. The photon which survives the termination is continued with its weight increased by a factor of $\mathcal{N}$.

With this method a slight change is required in the photon flux capture and the interpretation processes. Instead of keeping a count of the outgoing photons, a cumulative value of weight is maintained for each outgoing photon leaving the surface and on each interaction with a surface the photon's weight is scaled down by a factor equal to the reflectance of the surface. Thus in the computation of flux density the outgoing flux due to an emitter is given by the modified equation below:
Outgoing Flux due to each light source $=$

$$\frac{Cumulative\ weight\ of\ the Photons\ due\ to\ a\ Source}{Number\ of\ Photons\ Emitted\ from\ the\ Source} * Source\ Strength$$

The result of this technique for the test enclosure with 0.001 as threshold for cutoff and $\mathcal{N}$ as 2 has been shown in column III of Table.3. The results show a major improvement in the average photon 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 photon 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.3). The computational efforts have been measured by the number of ray-surface intersections. These results too show an improvement over the simple absorption method.

[3] New Webster Dictionary meaning: "A suicidal game or stunt in which the participants take turns spinning the cylinder of a revolver loaded with one bullet, placing the muzzle against the head and pulling the trigger".

We have incorporated both methods in our implementation. Either one may be chosen for any simulation run. Our experience so far has been that there is not much qualitative visual difference between the images created using simple photon absorption model and Russian Roulette model for equivalent computational efforts. This may perhaps be attributed to the fact that the total number of photons chosen for the simulation is sufficiently large so that the variance in the simple absorption method is kept low and does not result in any significant visual differences. We however do feel that as the complexity of the 3D configuration increases this variance reduction technique may yield better results for the same computational effort.

7 Conclusion

As with all Monte Carlo based methods the global illumination computed will fluctuate around the *real* value. The fluctuation can be reduced by increasing the number of photons used in the simulation. Thus it is not possible to ascribe a 100% confidence to the values that we obtain due to the stochastic uncertainty inherent in the methods. Although in principle given the necessary computational resources we can approach such confidence. 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 computable. There are no known methods to estimate such numerical uncertainties. In contrast statistical methods do exist for estimating the accuracy of the results. However, we have not as yet made any investigations in this regard.

With our implementation we have tracked around a million photons for the environment shown in Fig.4 which has 661 surfaces, small and large.[4] So far we have found that for the simple absorption model 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 photons is not at all prohibitive. For the given environment the average number of reflections that a photon undergoes is about 3.57. This means that the tracking of one million photons would require about three and a half million rays to be traced in the environment. By using a suitable acceleration technique this ray tracing can be generally contained on the average. In fact we

[4]Equivalently around 3000 (constant intensity) patches for obtaining a solution using radiosity method.

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 of the environmental configuration.

Monte Carlo simulation, thus, provides a very simple method for the computation of the global illumination in the three dimensional environment. As the mathematics describing it is not highly sophisticated, the method is quite straight forward to implement. Further it appears to have a definite advantage over the other analytical computation techniques when the behavioural complexity of the environment goes up beyond the ideal diffuse or ideal specular behaviour largely assumed in the analytical methods. Thus environments with complex light sources or unevenly illuminated large and curved surfaces or with participating media [14] can be treated with greater flexibility, simplicity and speed.

8 Acknowledgements

We are grateful to Dr S Ramani, Director, NCST for his support and encouragement. We thank Dr S Gopalsamy of NCST for clarifying some of the mathematical details in the derivation of position sampling equations. We are grateful to Dr Greg Ward of Lawrence Berkeley Lab for providing us access to the geometrical models which simplified our task of creating a reasonably representative and realistic 3D environment and Mr R Venkatesh of Industrial Design Centre, IIT for helping us photograph the images from the display screen.

References

[1] James Arvo and David Kirk. Particle transport and image synthesis. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4):63–66, Aug. 1990.

[2] Michael Cohen and Donald P. Greenberg. The hemi-cube: A radiosity solution for complex environments. *Computer Graphics (SIGGRAPH '85 Proceedings)*, 19(3):31–40, Aug. 1985.

[3] Michael Cohen, Donald P. Greenberg, Dave S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3):26–35, March 1986.

[4] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, Jan. 1986.

[5] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice, Second Edition.* Addison-Wesley, 1990.

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

[7] 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.

[8] Roy A. Hall. *Illumination and Color in Computer Generated Imagery.* Springer-Verlag, 1989.

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

[10] James T. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):143–150, Aug. 1986.

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

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

[13] 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.

[14] S. N. Pattanaik and S. P. Mudur. Computation of global illumination in a participating medium by monte carlo simulation. *The Journal of Visualisation and Computer Animation, To appear*, 4, 1993.

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

[16] Jerome Spanier and Ely M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems.* Addison-Wesley Publishing Company, 1969.

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

[18] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21(4):311–320, July 1987.

[19] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22(4):85–92, Aug. 1988.

[20] David R. Warn. Light control for synthetic images. *Computer Graphics (SIGGRAPH '83 Proceedings)*, 17(3):13–22, 1983.

Appendix : Sampling Methods

Here we discuss the various sampling methods used in our simulation work. Text books on Monte Carlo techniques [11, 16] have discussed the methods in greater detail.

.1 Sampling by Transformation of Random Variable

The basic idea behind this method is to provide a mechanism to transform a sample in the uniform random distribution and arrive at a random sample in the required distribution.

Principle : If x is a random variable with cumulative distribution function(*cdf*) $F_X(x)$ and probability distribution function(*pdf*) $f_X(x) = dF_X/dx$ and $y = y(x)$ is a continuous nondecreasing function of x then :

$$y(X) \leq y(x) \iff X \leq x$$

So

$$P\{y(X) = Y \leq y(x)\} = P\{X \leq x\}$$

So

$$F_Y(y) = F_X(x) \quad where \quad y = y(x)$$

So

$$\frac{dF_Y(y)}{dx} = f_X(x) \quad or \quad f_Y\frac{dy}{dx} = f_X(x) \quad or \quad f_Y = f_X(x)\frac{dy}{dx}^{-1}$$

This means given x, a random variable with *pdf* $f_X(x)$, and *y=y(x)*, the *pdf* of y, $f_Y(y)$ is $f_X(x)(dy/dx)^{-1}$.

Method : Given is the problem of sampling *pdf* $f_Y(y)$. Let us assume that y is some nondecreasing function of the uniform random distribution x. So from the above principle the *cdf* of both the distributions are related by the equation :

$$F_Y(y) = F_X(x)$$

The *cdf* of the uniform random variable is

$$F_X(x) = \begin{cases} 0, & \forall x < 0; \\ x, & \forall 0 \leq x \leq 1; \\ 1, & \forall x > 1. \end{cases}$$

So in the range $0 \leq x \leq 1$, $F_Y(y) = x$. Solving this equation one can get the expression of y in terms of x. The sampling strategy is to compute a uniform random sample ξ_x and compute ξ_y by using the equation *y=y(x)*.

.2 Rejection Sampling

A trial value for a random variable is selected and proposed. This value is subjected to one or more tests on the basis of which, the value may be accepted or rejected. If it is rejected, the cycle of choosing and testing a trial value is repeated until an acceptance takes place.

Method : Let *f(x)* be the bounded density function and assume that it vanishes outside the interval *(a,b)*. We wish to construct samples from the density *f(x)*. Let M = sup $f(x)$ for $a \leq x \leq b$ and define $f_1(x) = f(x)/M$. So $0 \leq f_1(x) \leq 1 \ for \ a \leq x \leq b$. Generate a pair of random numbers (R_1, R_2) in the range of 0 to 1 and interpret $(a + R_1(b-a), R_2)$ as a point in the rectangle with base (b-a) and height 1. If this point falls below the graph of $f_1(x)$, i.e. if $R_2 < f_1(a + R_1(b-a))$, then $a + R_1(b-a)$ is accepted as a sample from *f(x)*. If not the process is repeated until a sample is accepted.

.3 Discrete Distribution Sampling

If we have a set of n events where each event k is occurring with a probability f_k and the task is to choose an event at random, then the sampling method is as follows:

> Generate a uniform random number ξ. Find the smallest l such that
>
> $$\sum_{k=0}^{l-1} f_k < \xi \leq \sum_{k=0}^{l} f_k$$
>
> where for $l = 0$ the sum is zero. This l is the required sample.