Contents

Chapter 1

Introduction

Today computers are being increasingly used to model complex three dimensional environments in applications such as computer aided engineering and architectural design, computer animation and virtual environments[22, 23, 44, 51]. The modelled three dimensional configurations are then visualised by rendering images of these environments as seen from different view points. Over the last three decades rendering techniques have been continuously evolving to greater levels of sophistication in terms of the complexity of environments and the realism with which the images are produced. In the beginning it was hidden line drawings. The advent and wide spread use of raster display technology led to the development of techniques for producing colour shaded images. With rapid advances in high speed computing the emphasis today is on the synthesis of realistic images[25].

In all image synthesis techniques the fundamental step is computation of the amount and nature of the light from the three dimensional environment reaching the eye in any given direction[29]. Computer graphics rendering techniques carry out this computation by simulating the behaviour of light in the environment. Greater degrees of realism would mean higher correlation between the simulation and the physical world. In the physical world lighting, reflection and scattering effects are very complicated and subtle. Every object receives light directly from light sources, or indirectly from reflection or scattering by other neighbouring objects. For realistic image synthesis these intra-environmental effects must be modelled in great detail. In computer graphics the indirect lighting is often called as *global illumination.*

Light is a form of radiant energy and its behaviour has been extensively studied and mathematically modelled to a very high degree of sophistication in other disciplines such as radiative heat transfer[64] and neutron transport[43]. The prime problem that computer graphics addresses is the derivation of computationally tractable algorithms for carrying out the simulation based on these mathematical models.

This thesis presents the results of a detailed investigation of illumination computation and rendering techniques. From a theoretical point of view the primary contribution is the development of a mathematical framework of adjoint equations which provides the basis for all known illumination computation techniques. This mathematical framework consists of two integral equations - the *radiance* and the *potential* equation, which are duals of each other. While the radiance equation has been known in one form or the other to computer graphics community, the potential equation for illumination has been introduced for the first time in this thesis. The significance and importance of this new mathematical framework stems from the fact that it not only enables us to review and analyse existing methods but also provides the necessary handles for deriving new and efficient algorithms for simulating the behaviour of light in a manner closely correlating to the physical world.

On the practical side we describe new algorithms that simulate the particle model of light using Monte Carlo methods. The algorithmic improvements made possible by the use of the mathematical framework of adjoint equations are then demonstrated. Compared to previous work these algorithms can handle more general and complex environments. We also present the results of a straight forward implementation of these algorithms showing that these algorithms are computationally tractable.

1.1 Modelling Light Behaviour: The first step

Even though the behaviour of light, the optical properties of solid and non-solid material and the interaction of light with these materials have been extensively studied in physics, the mathematical models used are not easily accessible to computer graphics people. It becomes essential to glean through these physical models and extract and reframe these mathematical derivations to the extent that they become amenable to algorithmic simulations. Early image synthesis algorithms resorted to the use of very elegantly formulated empirical models which define the outgoing light in any direction as a function of the incoming light energy from all (i.e. global) directions. Due to the limited nature of computer processing power then available, the rendering algorithms applied these models locally around a surface without responding to the availability of global illumination in any significant fashion[29]. Subsequently there have been major shifts to the use of models that are derived from the physical behaviour of light. An in depth study of the development of the light behaviour models used in computer graphics has thus been the first step in our research.

As said earlier, light, a radiant form of energy originates at a light emitting source, and travels in the environment interacting with various places in the environment along its path. These interactions take one of two forms:

1. scattering, in which there is a change in travel path of the light at the point of interaction,
2. absorption, in which this radiant energy is ultimately lost.

An environment is composed of objects, defined in terms of their geometry and material properties, and a medium inside which the objects are embedded. The medium is composed of a variety of gaseous/floating material. Light can interact anywhere in the environment, in the medium, on the surface of an object or even inside the object. In the case of scattering, the point of interaction may be treated as an indirect source of light, as, for all practical purposes, when seen locally, light seems to come out of that point. In other words, the point of interaction can be said to be illuminated. The various scattering interactions taking place with the objects are generally classified as being reflections or refractions. The reflection term is used when the interacting light leaves a surface of the object from the incident side. If we define a sphere around any point of interaction, light exiting in any one or more directions in the hemisphere on the incident side is assumed to be due to reflection. Light exiting the other hemisphere is assumed to be due to refraction. Opaque objects do not refract light.

These different interactions result in some particular distribution of light in the environment. The illumination in the environment due to this distribution is global illumination. The global illumination problem can now be restated as follows:

Given

(i) the geometry of the objects,

(ii) optical properties of the objects and the medium,

(iii) a point of the environment, and

(iv) a direction

Compute

a measure of the exact amount of light leaving or reaching that point in that direction.

In most of the environments that we come across in real life the medium is non-interacting in nature. There could be occasional occurrences of localised volumes such as smoke and fire. Hence many of the illumination computation methods cater to environments with non-participating medium. In such a situation light travels in a straight path in the medium until it hits an object surface. Therefore it is only the object geometry and the object surface properties that influence the light distribution. Surface properties usually are made available in one of two forms:

1. As measured data in the form of reflection/refraction flux defined as a function of light incident from different directions[72]. (Flux is the energy propagating per unit time.)

2. As mathematical expressions defined in terms of various surface modelling parameters[3, 6, 17, 32, 37, 53, 54, 72].

In illumination computation methods the light metric commonly used is *radiance*. The basic reason is that the visual brightness of any point in the environment is proportional to the radiance leaving the point along the view direction[4]. Also this metric is dependent only on the direction and independent of position along the direction. Thus light interaction properties are often expressed in terms of outgoing radiance per unit incoming light flux. Such a property expressed for surface reflectance is termed as the bidirectional reflection distribution function, *brdf*[64].

Using the brdf of a surface point we can derive a mathematical expression for the outgoing radiance at that point as a function of the surface emission radiance and the incident radiance from all incoming hemispherical directions around that point. This equation, completely derived later in Chapter 2, we call as the *radiance equation*[1]. The problem of global illumination computation may then be comprehensively seen as solving the radiance equation for every point in the environment.

1.2 Solution Strategies: State of the Art

Right from the beginning the illumination computation problem has been and continues to be an interesting and challenging problem widely researched in the field of computer graphics. Literature abounds with any number of extensions to a few basic methods. A clear and comprehensive understanding of all these methods can be obtained by using the radiance equation as the underlying mathematical basis for carrying out the simulation.

The radiance equation is a complex integral equation. In general there does not exist any closed form solution for such an equation. Hence most of the illumination computation methods are basically approximate solution methods derived under different simplifying assumptions. The often used simplifying assumptions are as follows:

(i) the light sources are point light sources,

(ii) surfaces are uniformly diffuse or perfect mirror reflectors,

(iii) inter-reflections are not significant,

(iv) surface geometries are generally simple.

As we reduce the number of these simplifying assumptions, the computational methods get more complex and also more expensive. The development of computationally tractable methods for dealing with more and more general environments continues to

[1] The rendering equation given by Kajiya[38] is the one most often referred for illumination computation in Computer Graphics. It defines a similar relationship but in terms of points rather than directions.

be a very challenging problem.

A review of the existing computational methods show that they follow one of two basic strategies. They are as follows:

1. The Light Gathering Strategy: In this, illumination at a point is computed by gathering all the light incoming from the immediate surroundings of the point.

2. The Light Shooting Strategy: This strategy simulates the natural light propagation process. Starting from the light sources, light is distributed into the immediate surroundings of this point of origin. This process is continued until finally all the light reaching the point of interest has been computed.

The radiance equation is very natural for use as the mathematical basis for the methods based on the gathering strategy. Though not as natural it must be recognised that the methods based on the shooting strategy can also be derived from the radiance equation. This has to be so because both methods compute illumination of a point in an environment and it is precisely for this that the radiance equation provides a mathematical expression.

A secondary categorisation of illumination computation methods can be obtained from the observation that any numerical computation method for solving the radiance type of equation can follow either a deterministic or a nondeterministic approach. Deterministic methods are usually more efficient but applicable to somewhat restricted environments and limited in the global illumination that they use. The basic ray tracing method[75] which is particularly appropriate for specular environments and the standard radiosity method[27] which is most suitable for largely ideal diffuse environments are the most popular methods using this approach. On the other hand nondeterministic methods do not need many simplifications and hence provide easy extensibility to more general environments. Distribution ray tracing, path tracing and Monte Carlo based methods are typical examples[16, 14, 38, 62, 63]. However, the results obtained using such methods often tend to have high variance associated with them. One may have to resort to complex variance reduction techniques to reduce the variance[40].

A majority of the existing computational methods, using both deterministic and nondeterministic approaches are based on the gathering strategy. A few that are based on the shooting strategy mainly use the deterministic approach. Most notable of these is the progressive radiosity method[12] in which light from light sources and other bright surfaces is progressively shot to all other surfaces in the environment. Tracing light rays, also sometimes called as forward ray tracing[10], is another deterministic method based on the shooting strategy. Generally speaking these methods have received much less attention than methods based on the gathering strategy. The prime reason put across by many is the following:

> Illumination computation in Computer Graphics has always been very closely coupled to image rendering. Images of environments are usually required from specific view points. It "looks" much more efficient and sensible to gather illumination at the view point and not bother too much about computing illumination for the rest of the environment.

The basic ray tracing techniques therefore produce only view dependent illumination information. The standard radiosity method does produce view independent illumination information but is then restricted to strictly ideal diffuse environments. While some methods have been proposed for extending the standard radiosity method to produce view independent illumination information in more general environments these have yet to find real application.

If we look at methods based on the shooting strategy, at first sight, at least, it does appear that the process may result in unnecessary illumination computations by shooting light even into regions which are in no way connected with the view point. Though, going strictly by the definition of global illumination, every point of the environment contributes to the illumination of another point and is equally important. All the same, most interestingly, progressive radiosity and its derivatives are the state of the art techniques for global illumination computation.

Progressive radiosity uses the deterministic approach. To the best of our knowledge there have been no serious efforts earlier towards the development of a method that uses a non-deterministic approach based on the shooting strategy. As mentioned earlier without the use of variance reduction techniques such an approach may be computationally intractable except for simple environments. A sound mathematical basis is a must if efficiency improvements have to be incorporated into a non-deterministic light shooting method. The *potential equation* for illumination derived for the first time in this thesis provides such a basis for all the shooting strategy methods, deterministic and non-deterministic and enables performance/efficiency improvements to be made to algorithms using this strategy. The potential equation is the dual of the radiance equation and along with the radiance equation forms an adjoint system of equations that provides the basis for all known illumination computation methods.

1.3 Non Deterministic Particle Tracing : A New Algorithm

The particle model of light is a natural choice for simulating light behaviour using a non-deterministic method based on the light shooting strategy. In this model particles, packets of light energy, are emitted from the light emitting source in different directions. A particle on collision with the objects in the environment either looses its energy completely or gets rebounded (reflected) or refracted and changes its direction of propagation. The reflected/refracted/emitted particle flux given by the number of particles per unit time is a measure of the illumination of points in the environment. Monte Carlo methods[31, 40, 56, 68] can be used to simulate this particle model and obtain an estimate of the global illumination. In this thesis this simulation has been termed as *particle tracing*.

In particle tracing the emission and interactions all take place in a nondeterministic fashion by random sampling unique probability distribution functions. For example: for the emission of particles from an emitter of uniform strength the positions of the particles are chosen in such a way that after a reasonable number of emissions the particle density over the surface is almost constant. The path of each particle emitted is traced by following it along the emission direction to find the surface of interaction and choosing the type of interaction (reflection, refraction or absorption). If the particle is not to be absorbed then its path along its reflection or refraction direction is followed and the process continued until the particle is eventually absorbed at some surface.

On simulating the behaviour of a sufficiently large number of particles the actual number of particles coming out of a surface provides an estimate of the particle flux at the surface. As the simulation progresses, with increase in the number of particles this estimate is progressively refined. We have carried out a detailed comparative performance analysis with a straight forward implementation of standard radiosity to show that the estimates are equally good and the computation times are comparable. We have also carried out extensive experiments to see its usefulness in the presence of complex environmental behaviour such as complex geometry and complex emission and reflection behaviour and obtained encouraging results. As the demands on the accuracy of the illumination computations increase the total number of particles whose behaviour has to be simulated increases rapidly. The use of variance reduction techniques becomes mandatory. Such techniques have been extensively studied in the applications of Monte Carlo methods in other disciplines[35, 43]. Absorption suppression is one such technique. In this, some of the particles are not absorbed but are allowed instead to continue their flight in the environment though with reduced illumination capacities. The use of this technique does result in some efficiency improvements but cannot be considered as being substantive.

1.4 Particle Tracing in Participating Volumes : Simulation in Complex Environments

The particle tracing method briefly described above considers the interaction of light with surfaces only. An implicit assumption is that the ambient medium, generally clear air, does not in any way affect the flight of the particles. When there is smoke, or dust, or water vapour in the air this assumption is no longer valid. These materials participate in the light propagation process and may emit light particles, absorb and/or scatter particles making themselves visible. Accounting properly for such participating volumes during illumination computation is in general very complex and also computationally expensive. There have been some extensions to the ray tracing[39, 42, 48] and standard radiosity methods[58]. But overall the complexity involved has been such that real application of these methods has not yet taken place.

Fortunately for us the non-deterministic particle tracing method is inherently capable of handling more general environments and is also comparatively easier to extend. Basically we have to compute a potential point of interaction in the participating volume whenever a particle is traveling through it. The procedure for computing this point of interaction has been derived from the physics of light-volume interaction. Using this procedure we have extended the basic particle tracing algorithm to efficiently carry out the particle tracing in the the presence of participating volumes. These have been implemented and tested using test environments created by using simple volumetric modelling techniques. Once again for performance efficiency improvements standard variance reduction techniques like forced interaction have been incorporated. In forced interaction a particle is forced to interact with all the participating volume sub-elements encountered in its path. The extension is straightforward and illumination results with reasonable accuracy have been obtained for participating volume environments of moderate complexity. Overall however one comes to the conclusion that with increasing demands on the accuracy of illumination computations much too many particles are needed for the simulation of complex environment.

For improvements in the efficiency of the particle tracing algorithm two aspects have to be considered.

1. The particle behaviour simulated accurately reflects the physical model.

2. In such a simulation many particle traces may be unnecessary. That is, they may not contribute in any significant manner to the illumination computation.

We must evolve a method which without loosing the correlation with the physical model traces particles in such a way that they all make significant contributions to the global illumination computation. For this a sound mathematical basis has to be the foundation.

1.5 The Potential Equation : Mathematics for Performance Improvements

Analysis of a particle trace in the particle tracing process shows that the particle can be assumed to take a random walk among a number of possible steps with each step chosen from its ancestor by some random sampling process. Absorption being one of the many possible steps, the random walk is eventually terminated when the particle is absorbed. Random walk methods are known to be used for solving integral equations[56]. Already its use in solving the rendering equation has been propagated in Computer Graphics by the name of *path tracing*[38]. Surely the particle tracing process is also solving some integral equation. Using this insight the potential equation has been formulated. The *potential equation* is an integral equation similar in nature to the radiance equation. The name *potential* has been chosen because the equation expresses the potential capability of every point in the environment towards the illumination of a given point or a given region in the same environment. All the shooting based methods including progressive radiosity and particle tracing are basically different methods of solving the potential equation. The potential equation and the radiance equation are duals in the sense that they both can define the same illumination quantities. Together they form an adjoint system of equations for solving global illumination. It is thus conjectured that any global illumination method is a solution method for either of these two equations or a combination of these two.

It is known that the efficiency of random walk based solutions can be increased by careful transformation of the underlying mathematical descriptions defining the starting state and the transition functions to move from one state to another. Most important among such transformations is importance biasing and the estimation process known by the name of importance sampling. This concept is finally used in this thesis and a highly improved and efficient particle tracing algorithm has been devised and implemented.

To sum up:

> Our main thesis is that global illumination information which is needed for realistic visualisation of complex three dimensional environments can be effectively computed using the new non-deterministic particle tracing class of algorithms, the mathematical underpinning for which is provided by the potential equation. This new equation along with the earlier known radiance equation forms an adjoint system of equations, a unified mathematical framework that provides the foundation for all global illumination computation methods and also includes the necessary mathematical handles for further development of efficient methods.

1.6 Organisation of this Thesis

In Chapter 2, the physics of light, its interaction with the environment, and various empirical and physical models of light-environment interaction are presented in a manner palatable to a computer graphics reader. The radiance equation, expressing the radiance of a surface point in terms of the host surface reflectance and the radiance of the surrounding points, is then derived as an integral equation. Lastly the general radiance equation that takes into account participating volumes is formulated.

Chapter 3 is a comprehensive review of the current state of the art in illumination computation. It shows how most of these light behaviour simulation techniques can be seen as different methods of solving the radiance equation. The two basic simulation strategies, gathering and shooting, are discussed. Within each strategy, methods are further categorised based on whether they use a deterministic or a non-deterministic approach. All methods based on the gathering strategy are directly seen as solutions of the radiance equation while methods based on the shooting strategy can only be indirectly derived from this equation. The chapter concludes with the observation that illumination computation by the use of non-deterministic methods based on the shooting strategy had not been throughly explored.

The particle model of light is the most natural candidate for simulation by a method which uses the non-deterministic approach and is based on the shooting strategy. Chapter 4 introduces the first in a class of particle tracing algorithms which simulate the particle model of light to compute global illumination in a three dimensional environment. Monte Carlo basics and the necessary Monte Carlo sampling techniques needed in this simulation are also presented. The results obtained from an implementation of this algorithm are shown and its performance is compared with a simple standard radiosity implementation. The truly progressive nature of the algorithm is illustrated with the help of examples which show how the illumination computations get progressively refined as the simulation proceeds. The algorithmic modifications necessary to handle complex surface geometry and more complex surface emission and reflection behaviour are presented and the various issues relating to rendering the image from the computed illumination are discussed. Finally the variance reduction technique of absorption suppression is used to improve the efficiency of the particle tracing algorithm.

The particle tracing algorithm is inherently capable of being extended, comparatively easily, to more general environments. This is demonstrated in Chapter 5 by showing the changes necessary to handle participating volumes. First the algorithms to efficiently sample the interaction point and trace the particle in the participating volume are presented. Next various modelling techniques to model volume elements have been proposed and used for creating representative test environments with participating volumes. A method for image rendering in the presence of participating volumes is also discussed. Finally implementation of variance reduction techniques such as absorption suppression, forced collision, and particle divergence and their effects have been analysed.

The primary theoretical contribution of this thesis, the potential equation and the mathematical framework of adjoint illumination equations is the topic of Chapter 6. Using intuitive concepts the potential equation is derived and its duality with the radiance equation is proved. That all shooting strategy methods, including progressive radiosity and particle tracing, can be naturally derived as solutions to this potential equation is shown next. The efficiency of the particle tracing algorithms can be increased by the use of suitably transformed probability distribution functions. A number of biasing schemes are devised for this purpose. Implementation issues are discussed and the performance improvements obtained from a simple straight forward implementation are presented.

Chapter 7 is the concluding chapter. The results of the research are analysed along a number of important dimensions such as environmental complexity, image rendering, implementation considerations and relationship with other work. Possible extensions to the method and potential avenues for future research are also briefly discussed.

Computation and measurement in any discipline require a thorough knowledge of the metrics involved. Appendix A, has been devoted to a brief discussion of various light metrics. Light being a form of radiant energy, the discussion starts with metrics used in radiometry, i.e. the measurement of radiant energy. Light is that portion of the vast spectrum of the electromagnetic radiation which generates a physical sensation in the eye. Hence light is also known as visible radiation. The extent of this sensation is dependent on the nature and the amount of visible radiation energy impinging on the eye. The slightly different but related set of metrics that are used for photometry i.e. the measurement of the light based on the visual responses, are then presented. Lastly the highly intuitive term brightness is formally related to *radiance/luminance*, luminance being the photometric equivalent of radiance.

Chapter 2

Light and its Interaction with the Environment

The appearance of objects in an environment is determined by the amount of light coming out of that object and reaching the eye or any other optical detector. In a general environment this light would be due to an object emitting, transmitting and/or reflecting light energy from its surfaces. The manner in which light is emitted, transmitted or reflected depends very much on the material, the geometry and other surface properties. Also light coming out from an object may interact with the medium in the environment before reaching the eye or the detector. This interaction is in the form of absorption or scattering of light by the medium. It is therefore of immense interest to understand the nature of light, its properties and interaction with objects and other matter in the environment before one embarks upon the study and development of methods for computing illumination.

2.1 Physical Model of Light

Electromagnetic Theory of Light : Light is basically flow of radiant energy which is capable of exciting the retina of the eye to produce a visual sensation. The flow of energy is in the form of electromagnetic waves[24, 60, 64]. Because of this wave nature, light is characterised by the wavelength of the corresponding waves. The common unit of wavelength is nanometer(nm). The range of electro-magnetic waves which constitutes light is approximately 380nm to 770nm. The different wavelengths of light generate the sensation of different colours in our eye. Wavelengths for different colours are approximately as follows: 380 to 450nm (Violet), 450 to 500nm (Blue), 500 to 550nm (Green), 550 to 600nm(Yellow), 600 to 650nm (Orange) and 650 to 770nm (Red).

Quantum Theory of Light : Despite of the wave nature of radiation it has some properties similar to that of a particle. In particular the energy carried by this electromagnetic wave at any frequency is an integral multiple of a quantum of energy specific to the corresponding frequency[24, 60]. This quantum of energy is called a photon and has a magnitude equal to $h\nu$, where h is Planck's constant and ν is the frequency of the electromagnetic wave. Using this model one can completely ignore the wave nature of the light and consider light to be flow of photons. On interacting with the surface of an object or with the medium an integral number of photons are reflected/scattered and the rest of the photons are absorbed.

In the study of light either of the models, the wave model or the particle model, may be used depending on whichever one is more suitable for dealing with the behavioural aspect that is of interest. Independent of the wave or particle nature of light, the following assumptions hold true.

1. Light travels from one point to another in a straight line.
2. Out of all possible paths that one might take to get from one point to another, light takes the path which requires the shortest time. This assumption is called the *Principle of Least Time.*

2.2 Radiance: The Metric of Light Measurement

The rate of flow of radiant energy is termed *radiant flux.* Radiance[1] is a measure of the radiant flux leaving a surface point in any direction and is defined as the flux per unit projected surface area normal to that direction and per unit elemental solid

[1] For more details on flux, radiance, its relationship to the object brightness and other light metrics the reader is referred to Appendix A.

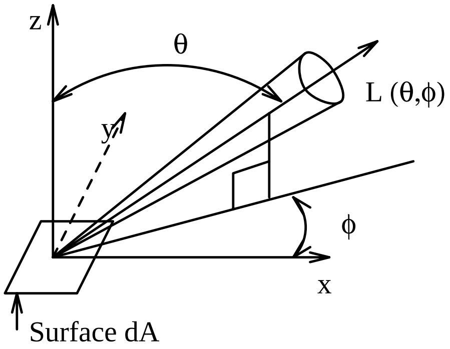

Figure 2.1: Radiance geometry.

angle, centered around the direction. Radiance is a convenient quantity for use in all illumination computations. From its definition, the radiance from a surface in a given direction in a noninteracting and nonemitting medium is constant for all positions along that direction. Also it can be shown[4] that the radiance of an object is a measure of the brightness of the object independent of the object's size and distance.

This invariance of radiance with distance in a nonparticipating medium also makes it a convenient metric to specify the magnitude of the interaction or emission when the medium is of participating nature. These effects can now be given directly as the change in radiance with distance. In the case of an interacting medium, the radiance is considered in terms of a local area within the medium. The projected area is formed by taking the area that the flux is passing through and projecting it normal to the direction of travel. The unit elemental solid angle is centered around the direction of travel and has its origin at the area of interest(Fig.2.1).

2.3 Light in a Nonparticipating Medium

An environment is assumed to be composed of solid objects and its encompassing medium. The medium may or may not be participating. By participating we mean the medium may contain gaseous or floating material which also interacts with the light eg. dust, smoke or some luminescent gas. First we shall discuss the interaction of light with objects in a nonparticipating environment. In such an environment, light travels between the surfaces in a straight line with no attenuation in the energy due to the medium. Light interacts with the surface of the solid object that it hits. This interaction is in the form of absorption, reflection and refraction. For surfaces of opaque objects absorption and reflection are the only interactions of relevance. These will be discussed in detail below. The quantity of light that is not reflected or refracted will be assumed to be lost by absorption.

2.3.1 Reflection

Reflection is a general term[1] used for denoting the process by which the incident flux leaves a stationary surface or medium from the incident side, without any change in frequency. On reflection from a perfectly smooth surface, the reflected flux leaves in a mirror direction which is uniquely characterised by the incident direction and the surface normal at the point of incidence as follows(Fig.2.2):
(1) the incident angle (the angle between the incident beam and the normal to the surface) is equal to the reflected angle (the angle between the reflected beam and the normal to the surface),
(2) the incident line, reflected line and the normal to the point where the reflection takes place are all in one plane.
This is the *law of reflection*. Surfaces with reflection properties satisfying this law are called as *specular* surfaces, and such reflection is called as *regular* or *specular* reflection.

When light goes from one medium to another it does not go in a straight line. At the interface it bends. The light is said to refract(Fig.2.2). The extent of refraction is given by the expression "η_1 sin(angle of incidence)=η_2 sin(angle of refraction)" where η_1 and η_2 are respectively the refractive indices of the first and the second medium, and are the ratios of the velocity of light in vacuum to that in the medium. This is the *law of refraction* and the equation is *Snell's Law*.

When light hits the surface of an object, none, some or all of the light may be reflected and/or refracted from the material. If all of the light is reflected and/or refracted then the object is said to be non-attenuating, otherwise it is said to be attenuating. For reflection from the surfaces of a nonattenuating object, the fraction

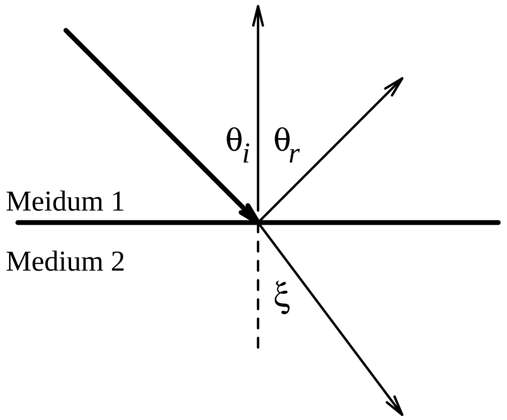

Figure 2.2: Reflection and refraction geometry.

of the light flux reflected, $\mathcal{F}$, is given by[64] Eq. 2.1 below.

$$\mathcal{F}(\theta_i) = \frac{1}{2}\left[\frac{(\eta_2\cos\theta_i - \eta_1\cos\xi)^2}{(\eta_2\cos\theta_i + \eta_1\cos\xi)^2} + \frac{(\eta_2\cos\xi - \eta_1\cos\theta_i)^2}{(\eta_2\cos\xi + \eta_1\cos\theta_i)^2}\right] \tag{2.1}$$

where θ_i is the angle of incidence and ξ is the angle of refraction. $\mathcal{F}$ is known as the Fresnel Coefficient. For reflection from the surfaces of attenuating material the expression for the Fresnel Coefficient is far more complex. When the incidence is from air or vacuum the expression is given by Eq. 2.2 [64, page 100] below

$$\mathcal{F}(\theta_i) = \frac{1}{2}\frac{a^2+b^2-2a\ \cos\theta_i + cos^2\theta_i}{a^2+b^2+2a\ \cos\theta_i + cos^2\theta_i}\left[1+\frac{a^2+b^2-2a\ \sin\theta_i tan\theta_i + sin^2\theta_i tan^2\theta_i}{a^2+b^2+2a\ \sin\theta_i tan\theta_i + sin^2\theta_i tan^2\theta_i}\right] \tag{2.2}$$

where

$$2a^2 = \sqrt{(\eta^2-\kappa^2-sin^2\theta_i)^2+4\eta^2\kappa^2} + (\eta^2-\kappa^2-sin^2\theta_i),$$
$$2b^2 = \sqrt{(\eta^2-\kappa^2-sin^2\theta_i)^2+4\eta^2\kappa^2} - (\eta^2-\kappa^2-sin^2\theta_i),$$

θ_i is the incident angle,

κ is the coefficient of absorption (*extinction coefficient*), and

η is the refractive index of the material.

Equations 2.1 and 2.2 above giving the fraction of reflected energy can only be used for objects with perfectly smooth surfaces. In practice, object surfaces are not perfectly smooth and hence do not exhibit specular reflection. That means the surface finish is such that light hitting the surface does not reflect exactly in the mirror directions. Rather it reflects in a range of directions. Such reflections are known[1] as diffuse reflections and corresponding surfaces are known as diffuse reflectors. Diffuse reflection may be further categorised into: narrow-angle diffuse, and wide-angle diffuse. In narrow-angle diffuse reflection, light flux is reflected at angles close to the direction which the light flux would take by specular reflection. While in wide-angle diffuse reflection, light flux is reflected at angles near and away from the specular reflection direction.

The light that is refracted at the surface of the object passes through the bulk of the object material. During this passage some of the light may be absorbed and some may be scattered. Depending on the composition of the material this absorption and scatter will vary from one point to another along its path inside the material. When modelling the optical behaviour of objects in Computer Graphics the major emphasis is on the interaction of light at the surface of the object. The phenomenon of reflection has thus received the maximum attention and highly sophisticated physical and mathematical models have been evolved for reflection. On the other hand, while the behaviour of light as it passes through optically thin gaseous material has been modelled to some extent, not much effort has been put so far into modelling the behaviour of refracted light as it passes through optically thicker liquid or solid material. There have been a few attempts reported in the literature and these are for the highly restricted situations listed below:

(i) objects of pure material like diamond[77],

(ii) objects of uniform material composition like glass[30, 75], and

(iii) very thin transparent objects like paper or cloth[57, 61], where the thickness of the material can be ignored.

In all these three cases the behaviour of light is modelled in two parts – reflected light and transmitted light. It is clear that in all the above cases light can be assumed to be transmitted either in an ideal diffuse manner or in a highly directional manner. As a result the same models as used for modelling ideal diffuse and specular reflection could be used for the transmitted light at the surface. However, more sophisticated models for refraction have yet to be evolved. In the rest of this chapter we shall discuss in detail various models for reflection. We shall also discuss the modelling of the behaviour of light as it passes through participating volumes.

2.3.2 Reflectance: The Measure of Reflection

The most commonly used measures of reflection are *reflectance*, ρ, and *bidirectional reflectance distribution function(brdf)*, f_r.

Reflectance is the ratio of reflected flux to the incident flux. Reflectance is a function of the spectral distribution characteristic of the incident flux and the geometry of the incident flux and the reflected flux. Depending on the geometry of the incoming and outgoing directions different reflectance terms are used. They are[1]: *bihemispherical*, *hemispherical-conical*, *hemispherical-directional*, *conical-hemispherical*, *biconical*, *conical-directional*, *directional-hemispherical*, *directional-conical*, *bidirectional* and *hemispherical* reflectances. All of these are defined as ratios of reflected to incident flux. Hemispherical refers to flux in a hemisphere of directions, conical refers to flux within a specified cone, and directional refers to a specific direction only. In this context the term directional is used for the incident flux when the incident flux is collimated, and is used for reflected flux when the the size of the solid angle of the collecting element is specified. If no qualifying geometric adjectives are used, reflectance for directional incidence and hemispherical collection is assumed. The Fresnel coefficient $\mathcal{F}$ defined in Section 2.3.1 provides an expression for the directional-hemispherical reflectance for the surface of a homogeneous material.

The *brdf*, f_r, represents distribution of light in the reflection directions. It is expressed as the fraction: the reflected radiance along an outgoing direction over the irradiance[2] from the incident direction. The expression for irradiance, $dE_i(\theta_i, \phi_i)$, at any point due to the incident radiance $L_i(\theta_i, \phi_i)$, is as follows:

$$dE_i(\theta_i, \phi_i) = L_i(\theta_i, \phi_i) \cos\theta_i d\omega_i$$

[2] See Appendix I for definitions of irradiance etc.

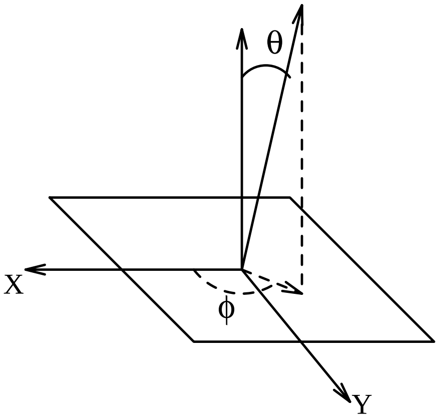

Figure 2.3: Representation of a direction.

where $d\omega_i$ is the differential solid angle around the incident direction.

At any surface point around the normal to the surface there is a hemisphere of directions for incoming and outgoing light. On this hemisphere, with respect to the normal at the point, let θ denote the cone angle and ϕ the circumferential angle where θ varies from 0 to $\pi/2$ and ϕ varies from 0 to 2π(Fig.2.3). Let (θ, ϕ) denote any direction[3]. Then f_r is given by the expression:

$$f_r(\lambda, \theta_r, \phi_r, \theta_i, \phi_i) = \frac{L_r(\lambda, \theta_r, \phi_r)}{dE_i(\lambda, \theta_i, \phi_i)} = \frac{L_r(\lambda, \theta_r, \phi_r)}{L_i(\lambda, \theta_i, \phi_i)\cos\theta_i d\omega_i} \tag{2.3}$$

where λ is the wavelength of the incident light,
(θ_i, ϕ_i) and (θ_r, ϕ_r) are respectively incident and reflected directions,
$d\omega_i$ is the differential solid angle around the direction (θ_i,ϕ_i),
$L_i(\lambda, \theta_i, \phi_i)$ is the radiance incident from (θ_i, ϕ_i) direction,
$L_r(\lambda, \theta_r, \phi_r)$, is the radiance reflected along (θ_r, ϕ_r) direction due to the incidence from (θ_i, ϕ_i) direction.

It is generally true that f_r is symmetric with respect to the reflection and incident direction. That means f_r is the same if we interchange the incident direction with the outgoing direction. Also for some of the surfaces f_r is independent of the reference axis for measuring the circumferential angle of the direction of the incoming light. Such surfaces are said to exhibit isotropic reflection behaviour. A special case of isotropic

[3] For conciseness instead of θ, ϕ we have often used Θ to represent the direction.

wide angle diffuse reflection, called *perfect diffuse* reflection, is one in which f_r is independent of incoming and outgoing directions. Such surfaces are also called Lambertian[4] surfaces.

2.3.3 Surface Reflectivity Models

In general it is difficult and also very expensive to measure the f_r for a surface as a function of every possible incoming and outgoing direction. Further, even if it were possible to exactly measure the values, storage requirements for this information would be prohibitive. To avoid these problems usually a mathematical model is used. This model is used to predict f_r for any incoming and outgoing direction. The mathematical model approximates the behaviour of a wide range of surface materials and surface finishes as a function of a few parameters.

The prime factor considered responsible for the hemispherical distribution of the reflected flux is the surface roughness. The roughness can be thought of as undulations on the surface with peaks and valleys which

- change the local surface normal at the point of incidence,
- affect the actual incidence at a point by shadowing,
- mask the actual reflection from a point.

These effects reduce the reflected flux in the specular direction and result in reflected flux in other directions as well. In some materials like non-metals light can penetrate some distance below the surface before getting absorbed. In such cases the multiple reflections on the layers just below the surfaces of the object, also contribute to the reflected flux in any hemispherical direction(Fig.2.4). The hemispherical reflection

[4] A Lambertian surface emits or reflects light in accordance with Lambert's cosine law. The Lambert's cosine law states that the luminous intensity in any direction from an element of a perfectly diffuse surface varies as the cosine of the angle between that direction and the perpendicular to the surface element. However, it must be noted that in many (older) Computer Graphics books, Lambert's cosine law has been stated to mean the relationship that the radiance from the reflecting surface is equal to the radiance of the point light source times the cosine of the angle made by the light source with the reflecting point.

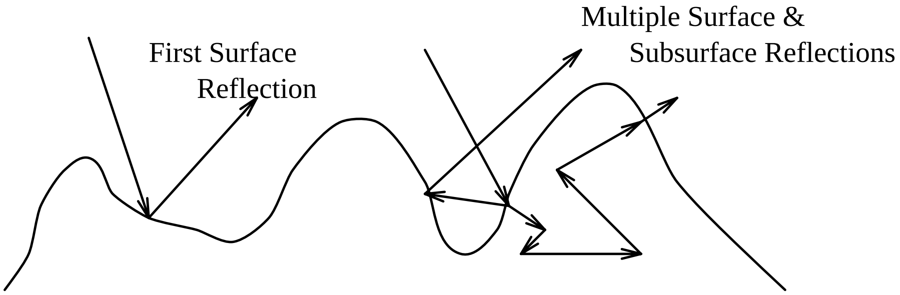

Figure 2.4: Reflection from an actual surface.

component resulting from the multiple reflections on the subsurface and multiple reflections inside the grooves of highly rough surfaces may be approximated as perfect diffuse reflection. Highly polished metals have very little of this kind of reflection. All of the outgoing illumination from a metallic surface is due only to the surface reflection. Hence the hemispherical reflectance distribution for the surfaces of metallic objects is attributed mainly to reflections (single or multiple) by the rough elements of the surface.

In general it is convenient to split f_r into a perfect diffuse, a directional diffuse and a specular component as follows(Fig:2.5):

$$f_r = f_{diffuse} + f_{directional} + f_{specular}$$

where $f_{diffuse}$, $f_{directional}$ and $f_{specular}$ represent the bidirectional reflection distribution functions for perfect diffuse, directional diffuse and specular reflection respectively. $f_{specular}$ is a delta function with nonzero values for the mirror direction and zero for all other directions. $f_{directional}$ results from reflection off the rough elements of the surface and in principle has nonzero values in all the hemispherical directions.

The functions $f_{specular}$ for mirror direction and $f_{directional}$ are simply given as some factor times the Fresnel coefficient, $\mathcal{F}$. This factor is the product of

1. the fraction of the effective surface area that receives flux from the incoming direction and reflects flux in the reflected direction, and

2. the fraction of the effective surface area whose local normal is such that the

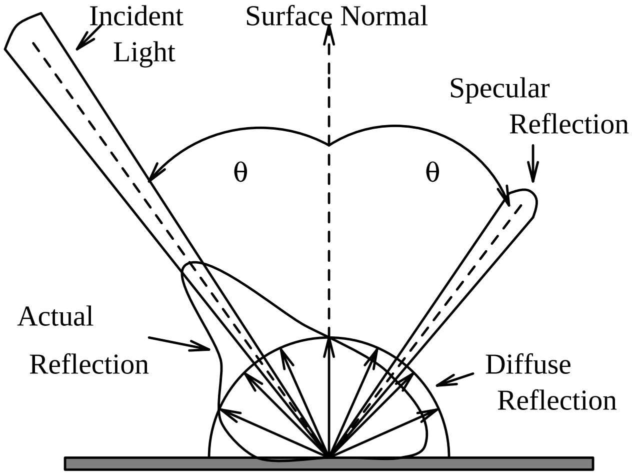

Figure 2.5: Reflection models.

> incoming direction, local normal and the reflected direction satisfy the law of reflection.

$f_{diffuse}$ is uniform in all the reflection directions and is represented by a constant value. This component inherently results from the very complex process of sub-surface penetration and multiple internal reflection of the light hitting the model. Its modelling is generally difficult. A simple method based on experimental evaluation is as follows:

> f_r controls the distribution of flux in the outgoing hemispherical direction for any incoming flux. An integration of this outgoing flux over the outgoing hemisphere will give the total reflected flux. This reflected flux can also be measured using an integrating reflectometer. The discrepancy between the measured value and the value predicted using only the directional and the specular components of the f_r provides an estimate for $f_{diffuse}$.

Various models for the analytical evaluation of $f_{directional}$ and $f_{specular}$ component of the *brdf* have been cited in the literature. Some are physically based[3, 6, 17, 32, 37, 54] and some are empirical[53, 72]. The physically based models try to model the roughness and derive expressions based on the material property using the laws of physics. Where as empirical models try to fit-in the experimental reflection data to some appropriate basis function, thereby giving a compact representation for the complex *brdf*. The basis functions chosen are such that they are computationally simpler and changes in their parameters create predictable variations in the surface reflection behaviour thus enabling the simulation of a wide range of surface behaviours. A few of the more well known physically based and empirical models are discussed below.

2.3.4 Physically Based Models

In these models, the physical characterisation of the roughness and the resulting reflections are most important. A parameter characterising the roughness effect is the *root_mean_square (rms) roughness*, σ. Roughness is relative to wavelength. A surface with some σ may be considered rough for light at smaller wavelength, whereas the same surface may be considered smooth for light at much larger wavelength. For that reason, often a derived parameter, $\frac{\sigma}{\lambda}$, called, *optical roughness*, is used. For any particular surface σ can be obtained by means of a profilometer[5] It must be kept in mind that σ does not give any information on the distribution of the size of roughness around the *rms* value. Ordinarily the roughness is very irregular, and statistical models are used to derive the distribution of the roughness. The often used distribution is Gaussian. Other important parameters characterising the roughness effect are the *auto correlation distance*, τ, and the *rms slope*, m(Fig.2.6). The *auto correlation distance*, τ, is a measure of the spacing between roughness peaks on the surfaces and *rms slope* is the root mean square slope of the undulations. Assuming isotropic reflection behaviour, $f_{directional}$ is expressed as:[6, 17, 32]

$$f_{directional} = \frac{\mathcal{F}}{\pi} \frac{G\,D}{\cos\theta_i \cos\theta_r} \tag{2.4}$$

where $\mathcal{F}$ is the Fresnel coefficient, G is the geometrical attenuation factor, D is the distribution of surface roughness, θ_i and θ_r are incident angle and reflection angle respectively. The expressions for G and D are functions of parameters $\frac{\rho}{\lambda}$, τ, m, the incident direction Θ_i, outgoing direction Θ_r and the physical properties of the material.

[5] A profilometer is an instrument that traverses a sharp stylus over the surface and reads out the root mean square vertical perturbation of the stylus.

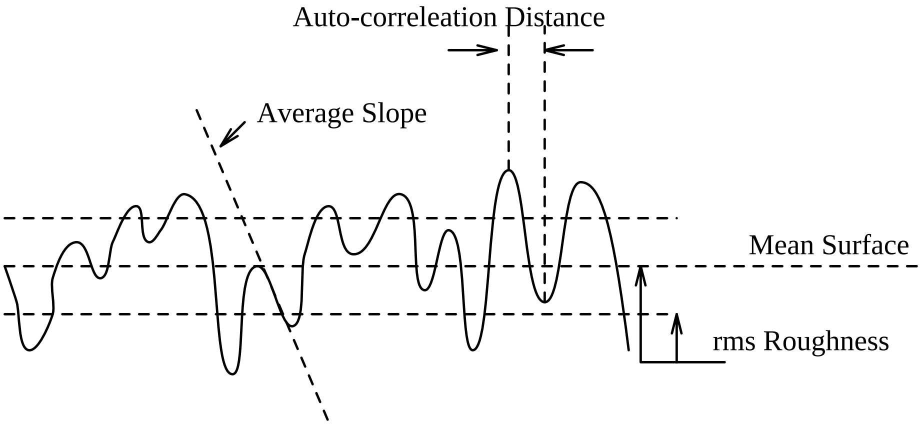

Figure 2.6: Roughness characterising parameters.

The actual equation for each of them differs from model to model depending on the approximations and the physical principles used to derive the equations.

Torrance-Sparrow Model : Torrance and Sparrow[6, 17] made a simplifying assumption that the surface may be assumed to be composed of a collection of specular micro facets which are oriented in random directions all over the surface and the microfacets exist in the form of $\vee$ shaped grooves. The component of the reflected light in any direction arises out of specular reflections occurring off the facets oriented in such a way that their specular reflection direction with respect to the incident light coincides with the direction of interest. Further these microfacets shadow and mask each other thus reducing the effective number of microfacets reflecting along Θ_r. Blinn[6] and Cook[17] provide a review of the various expressions derived for G and D using the Torrance-Sparrow Model.

Kirchoff's Tangent Plane Approximation : Kajiya[37] and He *et al*[32] have used Kirchoff's tangent plane approximation for the derivation of G and D factors for the Eq. 2.4. In this approximation, the electro-magnetic field at any point on the rough surface is approximated by the field which would occur if the surface were to be replaced by a tangent plane to the surface at that point. The formulations using this approximation are rather complex and can be found in [32]. He *et al*[32] state that their formulation for isotropic reflection behaviour compares favourably with experimental measurements of reflected radiation for a wide variety of metal, non-metal and plastic surfaces with varied roughness.

2.3.5 Empirical Models

As mentioned earlier empirical models are simpler to compute, intuitive and also provide a compact representation of the complex reflectance distribution. For the visualisation of a realistic environment it is often possible to use these models to predict the *brdf* values and avoid the rigours of the physically based models. We shall discuss two such models by Phong[53] and Ward[72] in this section. In both these models reflectance is assumed to be composed of a perfectly diffuse component and a directional component. The specular component is subsumed in the directional component.

Phong's Model : Phong's model[53] is one of the oldest and is a very simple empirical reflectance model. This model is the most often used reflectance model in image synthesis. This model gives an expression for bidirectional reflectance (not for *brdf*) to predict reflections from shiny surfaces. The expression is as follows:

$$\rho_{bd}(\Theta_r, \Theta_i) = k_d + k_s(\theta_i) cos^n \alpha \tag{2.5}$$

where

> k_d and k_s are called diffuse and specular reflection coefficients representing the fraction of perfectly diffuse reflection and directional reflection, and satisfy the condition $k_d + k_s \leq 1$,
>
> α is the angle that the direction Θ_r makes with the specular reflection direction of Θ_i(Fig.2.7) and
>
> n is an empirical roughness parameter controlling the rate of decrease in the reflected flux as a function of α. In the extreme as n tends to ∞ perfectly specular behaviour is modelled.

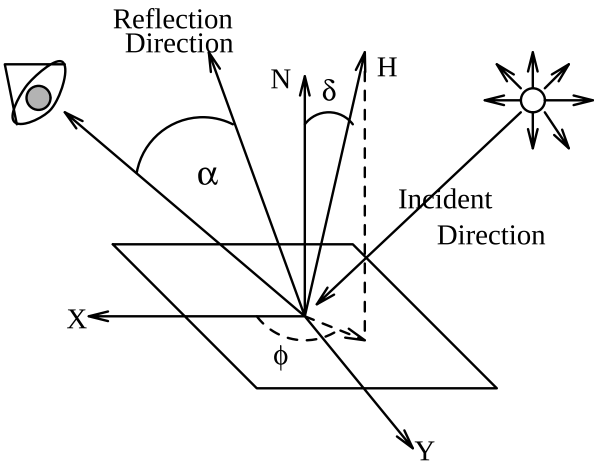

Figure 2.7: Phong's and Ward's reflection model geometry.

As can be seen from this equation, the model tries to fit a power cosine function to the reflection behaviour of shiny surfaces. The parameters k_s and n together express the directional reflection characteristics of a material. The parameter n may be thought of as a measure of surface roughness and the parameter k_s as a simple function of the Fresnel coefficient. In the original model no attempt has been made to derive the values for these parameters from physical principles. These numbers are empirically adjusted. For more reflective materials, the values of both k_s and n are larger. The range of k_s is between 10 and 80 percent and that of n is between 1 and 10. One disadvantage that must be noted is that no attempt has been made to conform to the law of conservation of energy. This means more energy may be reflected than it is incident. On the merit side associating Phong's model with a surface means choosing only two parameter values satisfying the above mentioned simple conditions. One can easily arrive at the required values by trial and error.

Ward's Reflection Model : Ward[72] has recently proposed an empirical model for anisotropic reflection. Though it is computationally more expensive than Phong's model, it has the following advantages:
(i) It fits the experimentally measured data for a wide range of surface reflectance data,
(ii) it has physically meaningful parameters, and

(iii) it satisfies the law of conservation of energy.

Further the equations proposed in this model adapt easily to Monte Carlo sampling[6] and hence are likely to be widely used in image synthesis.

The expression proposed for the *anisotropic brdf* ($f_{r,aniso}$) in this model is given by:

$$f_{r,aniso}(\Theta_i, \Theta_r) = \frac{\rho_d}{\pi} + \rho_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{e^{-tan^2\delta(cos^2\phi/\alpha_x^2 + sin^2\phi/\alpha_y^2)}}{4\pi\alpha_x\alpha_y} \tag{2.6}$$

where

ρ_d and ρ_s are diffuse and specular reflectance coefficients,

α_x and α_y denote the standard deviation of the *rms slope* in the **x** and **y** directions respectively,

δ is the angle between the average surface normal and the angle bisector of the incoming and reflecting direction, and

ϕ is the circumferential angle of this bisector(Fig.2.7).

For isotropic reflection $\alpha_x = \alpha_y = \alpha$ and hence the equation simplifies to

$$f_{r,iso}(\Theta_i, \Theta_r) = \frac{\rho_d}{\pi} + \rho_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{e^{-tan^2\delta/\alpha^2}}{4\pi\alpha^2} \tag{2.7}$$

The ρ terms in this equation may have spectral dependence. ρ_s may be computed using the Fresnel coefficient for the surface material. As long as the total reflectance $\rho_d + \rho_s$ is less than 1 and the α values are not too large, the above equations yield a physically valid function. Ward[72] has fitted experimentally measured data to this parametric equation and has tabulated ρ_d, ρ_s, α_x, α_y for a number of material/surface behaviours.

2.4 Emission

Light emission is the result of a process in which energy in some form, say chemical or electrical, is converted into electromagnetic radiation in the visible wavelengths. The emission results from the electronic transition of the molecules in the material. The radiation emitted from a solid actually originates within the solid. Like reflection,

[6] Detailed discussions on Monte Carlo Sampling can be found in Chapter 4.

emission results in a directional distribution of the emitted radiance. There has been very little attempt in the literature to model the light source. The often used emission distribution is the perfect diffuse or Lambertian emitter. Further the emitters are usually treated as time independent sources of constant emission flux.

2.5 Light in a Participating Medium

So far in our discussion we have assumed that the light when propagating from one surface to another either by emission or reflection is not attenuated. This is due to our assumption that interaction of the visible light with the medium through which it passes is negligible. However, if the medium interacts with the visible light, then the light coming out of a surface and the light reaching the surfaces surrounding that surface are no more the same. Depending on the medium through which the light has passed it is attenuated or augmented. Such media are said to be participating media. Attenuation is due to absorption and scattering, while augmentation could be due to emission in the medium or due to light scattered in from other directions.

2.5.1 Scattering

Scattering is a phenomenon which occurs when light strikes the particles present in the medium. On striking, some of the incident radiation may be reflected from the particle surface. This is termed scattering by reflection. The remaining portions of the radiation will penetrate into the particle, where part of the radiation can be absorbed, transmitted, or can undergo multiple internal reflection, and refraction. The redirection of light by these processes is called scattering by refraction. There is also scattering by diffraction. Diffraction is the result of a slight bending of the light propagation paths near the edge of the obstruction. For illumination computations the scattering process of interest may be categorised as:

- Isotropic scattering : Scattering is uniform in all directions. Like diffuse reflection this is basically the idealisation of the scattering process.

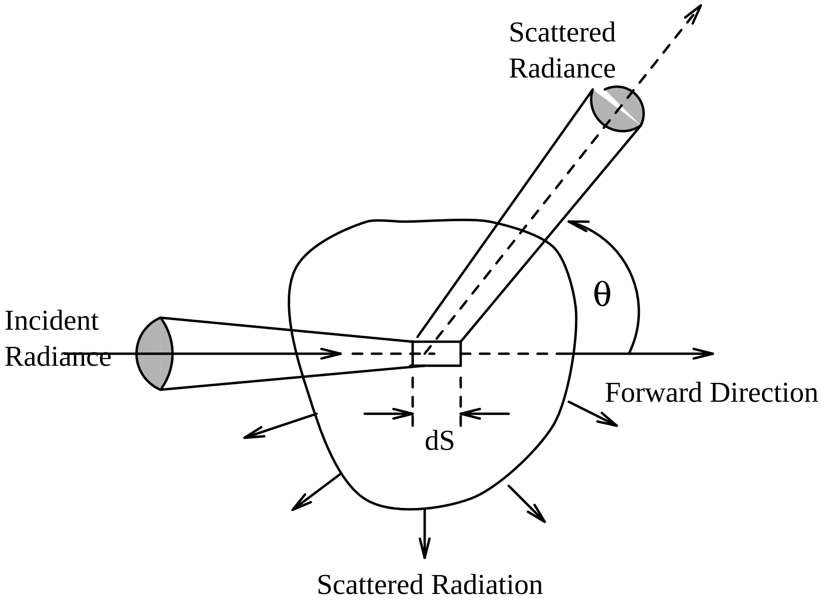

Figure 2.8: Scattering.

- Anisotropic scattering : There is a distribution of scattering directions. That means light is scattered nonuniformly in its surrounding.

It must be noted here that for the interaction at any point in the medium there is a complete sphere of incoming and outgoing directions to consider and not a hemisphere as was the case for reflection. So the direction is represented by θ,ϕ where θ takes a value from 0 to π and ϕ from 0 to 2π(Fig.2.8). A phase function,$\mathcal{P}(\theta,\phi)$, is used to describe the angular distribution of the scattered energy. The phase function gives the scattered radiance in a direction divided by the radiance that would be scattered in that direction if the scattering were isotropic. For isotropic scattering $\mathcal{P}(\theta,\phi) = 1$. Two of the models widely used for the phase function in anisotropic scattering are Rayleigh Scattering and Mie Scattering. In these models the phase function is independent of the circumferential angle ϕ.

Rayleigh Scattering : This is applicable in situations where the scattering particles are considerably smaller than the wavelength of light. The model predicts the phase function by the equation

$$\mathcal{P}(\theta) = \frac{3}{4}(1 + \cos^2\theta)$$

where θ is the angle the scatter direction makes with the direction of light incidence. Rayleigh scattering is applicable when the scattering is by gas molecules.

Mie Scattering : This is useful for explaining the directional distribution of the light scattered from the particles larger than the wavelength of light. By this model the phase function is represented by

$$\mathcal{P}(\theta) = \frac{3}{5}\left[(1 - \frac{1}{2}\cos\theta)^2 + (\cos\theta - \frac{1}{2})^2\right]$$

The general absorption and scattering behaviour of the medium is described by modelling the density distribution of the particles as a function of position, the reflection, refraction and absorption behaviour of each individual particle and an appropriate phase function. Computer graphics literature for this kind of modelling is rather limited[7, 41, 45, 48].

2.6 The Radiance Equation

Radiance from any surface point in a nonparticipating environment is due to reflection of incident radiation from the incoming hemisphere around the point and due to emission from that point. In this section we shall derive an expression for this outgoing direction. From the definition of *brdf* (Eq. 2.3) the measure of the reflected radiance as a function of the incident radiance from any direction can be written as follows:

$$L_o(\lambda, x, \Theta_o) = f_r(\lambda, x, \Theta_o, \Theta_i) L_i(\lambda, x, \Theta_i) \cos\theta_i d\omega_i$$

where

$L_i(\lambda, x, \Theta_i)$, $L_o(\lambda, x, \Theta_o)$ are respectively the incoming and outgoing radiance of wavelength λ at point x,
θ_i is the cone angle of the incoming direction,
$d\omega_i$ is the differential solid angle around the incoming direction.

Taking into account incidence from all the directions in the incoming hemisphere around the point x, the outgoing radiance due to reflection can be expressed as

$$L_o(\lambda, x, \Theta_o) = \int_{\Omega_x} f_r(\lambda, x, \Theta_o, \Theta_i) L_i(\lambda, x, \Theta_i) \cos\theta_i d\omega_i$$

where the integration range Ω_x denotes the incoming hemisphere around x.

Including the radiance due to emission we arrive at the general expression for the radiance from an opaque surface point in any nonparticipating environment.

$$L_o(\lambda, x, \Theta_o) = L_e(\lambda, x, \Theta_o) + \int_{\Omega_x} f_r(\lambda, x, \Theta_o, \Theta_i) L_i(\lambda, x, \Theta_i) \cos\theta_i d\omega_i$$

where L_e is the radiance due to the emission at point x. This equation is one of the fundamental equations used in our illumination computations. In an environment incoming radiance at x is due to the outgoing radiance at some point y visible to x along that direction. So we may rewrite the above equation as follows:

$$L_o(\lambda, x, \Theta_o) = L_e(\lambda, x, \Theta_o) + \int_{\Omega_x} f_r(\lambda, x, \Theta_o, \Theta_i) L_o(\lambda, y, \Theta_i) \cos\theta_i d\omega_i \tag{2.8}$$

where $L_o(\lambda, y, \Theta_i)$ is the outgoing radiance at point y visible to x along the direction Θ_i(Fig.2.9). Eq. 2.8 shall hence forth be referred to as the *Radiance Equation*. To the world of computer graphics Kajiya introduced a variant of the Radiance equation widely known as *Rendering Equation*[38] in the following form:

$$I(\lambda, x, x') = g(x, x') \left[E(\lambda, x, x') + \int_S \rho(\lambda, x, x', x'') I(\lambda, x', x'') dx'' \right] \tag{2.9}$$

where

$I(\lambda, x, x')$ is the intensity of light passing from point x' to point x(Fig.2.10) and is related to the outgoing radiance as $I(\lambda, x, x') = L_o(\lambda, x', \Theta_o) \cos\theta_o d\omega_o$

$g(x, x')$ is the visibility term which take a value of 1 if x' is visible to x, 0 otherwise,

$E(\lambda, x, x')$ is the emitted light intensity from x' to x, and is related to emitted radiance as $E(\lambda, x, x') = L_e(\lambda, x', \Theta_o) \cos\theta_o d\omega_o$.

$\rho(\lambda, x, x', x'')$ is the point to point transport reflectance and is related to f_r as $\rho(\lambda, x, x', x'') = f_r(\lambda, x, \Theta_o, \Theta_i) \cos\theta_i \cos\theta_o$, and

S, the range of integration, is the union of all the surface of the environment.

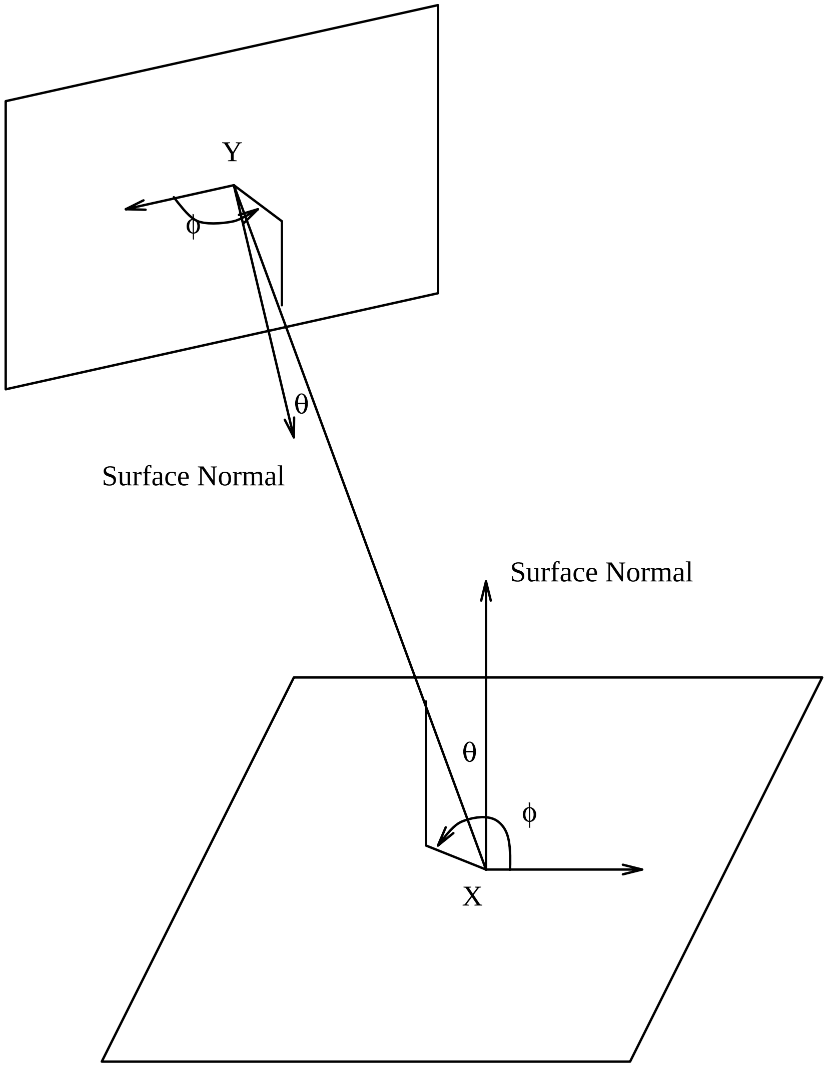

Figure 2.9: Radiance equation geometry.

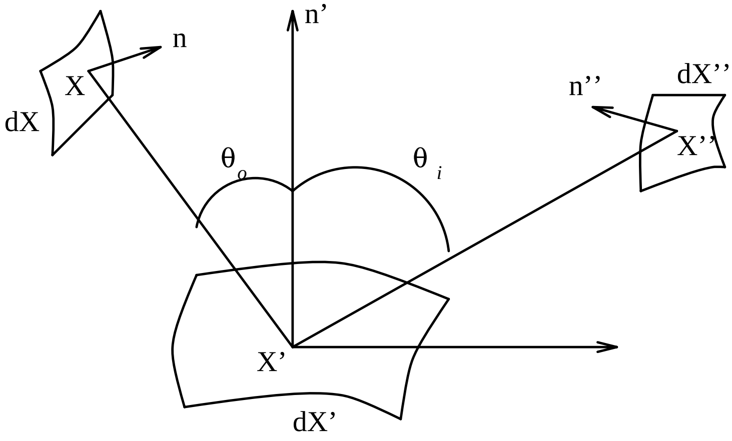

Figure 2.10: Three point geometry.

The equation states that the transport intensity of light from one surface point to another is simply the sum of the emitted light and the total light intensity which is scattered towards x from all other visible surface points.

2.6.1 Effect of Participating Medium

The radiance in a nonparticipating medium is invariant of distance along the direction of propagation of light. However in a participating medium various interactions that may occur along the path may cause its variation with distance. The phenomena like absorption, emission and scattering are responsible for this variation. Emission along the path may cause an increase in radiance. Absorption results in a complete loss of light and hence a decrease in radiance. Scattering distributes the light away from the direction of propagation and hence a loss in radiance. Further light moving in another direction may scatter into the direction of our interest and hence increase the radiance in that direction.

If we exclude the extraneous light coming in our direction of interest, i.e. the light due to scattering in and due to emission, and simply consider the fate of the light originating at a point in some surface area (could be hypothetical) then only two effects need be considered. They are the absorption losses and the scattering losses. It has been found experimentally that the resulting change in radiance depends on the magnitude of the local radiance.

$$dL = -K(s)Lds$$

Where K is the extinction coefficient of the medium in the differential layer and has a unit of reciprocal length.

$$K(s) = \sigma_a(s) + \sigma_s(s)$$

where σ_a is the absorption coefficient, and σ_s is the scattering coefficient[7].

The attenuation of light due to absorption and scattering along the path at a distance S can be derived as follows:

$$\begin{aligned} \frac{dL}{L} &= -K(s)ds \\ ln\frac{L(s)}{L(0)} &= -\int_0^S K(s)ds \\ L(s) &= L(0)e^{-\int_0^S K(s)ds} = L(0)e^{-\kappa_S} \end{aligned} \tag{2.10}$$

where $K(s)$ is a function of local parameters of the medium, and $\kappa_S = \int_0^S K(s)ds$, is the *opacity* or *optical thickness.* This equation is also known as *Bouguer's Law.*

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. If opacity $\gg 1$ then the medium is said to be optically thick, that means the mean penetration distance of light is quite small compared to the medium dimension. If opacity $\ll 1$ then the medium is said to be optically thin and the mean penetration distance of light is much larger than the medium dimension. Thus in an optically thin medium light can pass entirely through the material without significant absorption. For a medium of uniform composition $\kappa_S = \int_0^S K(s)ds = K\,S$. So $L(s) = L(0)e^{-K\,S}$.

[7] The symbol σ here should not be confused with the one used for *rms* roughness. Though a different symbol could be used in either place, we shall continue with this symbol as the literature usage of σ is common for both.

2.6.2 The Generalised Radiance Equation

Considering the change in radiance along the light propagation path in a participating medium, the general expression for the radiance, at any point x of the environment, in any direction Θ_o can be written as

$$L_o(\lambda, x, \Theta_o) = L_e(\lambda, x, \Theta_o) + \int_{\ominus_x} T(\lambda, x, \Theta_o, \Theta_i)\mathcal{L}_o(\lambda, y, \Theta_i) cos\theta_i d\omega_i \tag{2.11}$$

where

> x is any point in the environment (not restricted any more to be only surface points as in Eq. 2.8),
> L_o and L_e are respectively the outgoing and emitted radiances,
> the integration range $\ominus_x$ denotes the hemispherical directions for a surface point and denotes spherical directions for a volume point,
> T is f_r for a surface point and is $\sigma_s \mathcal{P}$ for a volume point,
> S denotes the distance of the nearest surface point along the direction Θ_i^{-1} from x and
> $\mathcal{L}_o(\lambda, y, \Theta_i)$ represents the cumulative outgoing radiance from every point, y, along the Θ_i^{-1} direction starting from x up to the nearest surface point along that direction and can be expressed as $\mathcal{L}_o(\lambda, y, \Theta_i) = \int_0^S L_o(\lambda, y, \Theta_i)e^{-\kappa_s} ds$.

2.7 Remarks

The study of the physical process of light energy interacting with matter is an important part of computer image synthesis as it forms the basis for the computation of colours in the synthesised picture. The computationally simpler models of the early days have all been replaced and augmented by more complex models gleaned from the work and literature of other physical disciplines. The formulation of the radiance equation and the generalised radiance equation (Eq. 2.8 and Eq. 2.11 above) is another step in the direction of assimilating such knowledge. These equations are required for the exact simulation of light interacting with complex object surfaces and participating media in a complex 3D environment. Though similar equations have been extensively studied by the discipline known as *Radiative Transport Theory*, the highly sophisticated mathematical methods are not easily accessible to computer graphics implementations. Also the complexity of the environments and the expected accuracy of the results differs. In computer graphics the prime interest in simulating the behaviour of light in an environment is to use the simulation results for image synthesis. Often, the geometrical complexity is very high, effects of participating volumes can be dealt with rather lower accuracy and simpler, less physically accurate reflectance models can be used, provided the computed images are almost realistic. In all cases, however, the radiance equations (Eq. 2.8 and Eq. 2.11 above) are fundamental and provide a basis for a complete simulation of the interaction of light in a complex environment with complex optical behaviour of the surfaces and the enclosing medium. In the following chapter, using these equations as the basis, we shall review well established illumination computation methods in the field of computer graphics.

Chapter 3

A Review of Illumination Computation Methods

The radiance equations, Eq. 2.8 and Eq. 2.11 presented in the previous chapter clearly show that the illumination at any point in a complex 3D 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. And hence the term global illumination. The radiance equation is a complex integral equation and computation of global illumination in any 3D environment would require the solution of this equation for that environment. Over the last few decades in the field of computer graphics we have seen an immense body of research carried out to compute illumination in an environment primarily for the purposes of image synthesis and rendering. Essentially all these methods can be seen to be providing some form of a solution to the radiance equation.

The complexity of the radiance equation is such that solving it completely and accurately for general environments is not computationally feasible. Hence most of the illumination computation techniques are arrived at after making some simplifying assumptions regarding the behaviour of the environment and also relaxing the accuracy to which illumination results are to be computed. Early illumination methods for example, considered only direct illumination from point light sources and approximated all indirect illumination by a constant term. In subsequent research efforts we therefore see a major thrust in the development of methods for capturing this indirect illumination. Considering the fact that image synthesis has been the primary motivation for computing global illumination, computational complexity is reduced in many of these by restricting the computation of illumination only to the visible points of the environment. Even for these visible points application of the complete illumination model is computationally expensive and is avoided whenever possible by interpolating from radiance values computed at a minimal set of discrete points in the environment.

Eq. 2.8 is an expression for the outgoing radiance from a point x on a surface along some direction Θ_x. Θ_x is a direction within the outgoing hemisphere erected at x with the normal to the surface at x. This means that for any given point and direction, say (x, Θ_x), solving of the radiance equation will provide us with a numeric value for the radiance, say $L(x, \Theta_x)$, from that point along that direction. If we make the assumption that the surface is diffusely reflecting (and diffusely emitting if the surface is also an emitter) then L is the same for all the outgoing directions. In such a case it is sufficient to solve the equation and obtain the value for radiance along any direction from that point. If we additionally assume that the radiance is uniform in the neighbourhood of the point, say in the small patch to which the point belongs, then we only have to solve the radiance equation for any point on the patch and along any direction. An illumination computation method provides a solution to the radiance equation at one or more points in the environment. The point(s) where the computation is to be carried out may be (i) predefined for the environment, for example as in the case of radiosity methods where the surfaces in the environment are discretised into small pathes and the solution is carried out so as to obtain radiance values at the centres of the patches or (ii) determined as the illumination computation progresses by the need to obtain radiance values at a selected set of primary points, usually belonging to the visible surfaces in the environment. If the solution at a point is to be taken as the representative of the solution of an area then the proper association between the point and the area is very important. This problem of suitably defining the points and the associated patches is generally known as the problem of discretisation of the environment[5, 9, 34, 47].

The radiance equation is a linear integral equation of the second kind[1]. A closed form solution for a general equation of this kind does not exist. However, one may resort to various numerical quadrature techniques to get a reasonably accurate solution for this integral equation. Such techniques may be broadly divided into deterministic and nondeterministic categories. Nondeterministic methods are generally based on principles of Monte Carlo quadrature. A method specific to the solution of integral equation of the second kind is the random walk. Monte Carlo quadrature and random walk techniques are briefly discussed in later sections. The choice of any particular method depends on its ability to handle a range of surface geometries, surface *brdf*s and surface emission properties. The integral term of the radiance equation is an expression that accounts for the light incoming from the hemispherical directions at the point and represents the process of gathering illumination information coming in from all directions of the hemisphere. As most of the methods are basically methods for solving this integral term we shall term them as *gathering methods*[2].

There are a few methods which compute global illumination but not by gathering. Rather they conform to the basic physical process of light propagation in which light originates from the source and interacts with the surfaces of the environment and gets distributed. We shall term the methods under this category as *shooting methods*. Later in this thesis, we shall show that the mathematical equation governing the basic shooting process is another equation which has been termed as the potential equation. Along with the radiance equation the potential equation forms an adjoint system of

[1] **Defns of Integral Equation of First & Second Kind[19]:** Let $K(s,t)$ be a continuous function of the two variables s and t defined over the domain $a \leq s \leq b$, $a \leq t \leq b$. Let $f(s)$ and $\phi(t)$ be two continuous functions of the variable s and t respectively defined over the interval $a \leq s, t \leq b$.

If the functions $f(s)$, $\phi(t)$ and $K(s,t)$ are connected by the equation $f(s) = \int K(s,t)\phi(t)dt$ then the equation is called a *linear integral equation of the first kind*.

If the functions $f(s)$, $\phi(t)$ and $K(s,t)$ are connected by the equation $f(s) = \phi(s) - \lambda \int K(s,t)\phi(t)dt$ where λ is a scalar parameter, then the equation is called a *linear integral equation of the second kind* with the kernel $K(s,t)$.

By these equations every continuous function ϕ is transformed into another continuous function f. The transformation is linear because the transformation of $c_1\phi_1 + c_2\phi_2$ results in $c_1 f_1 + c_2 f_2$. The primary interest in these types of equations is in determining $\phi(s)$ when $f(s)$ is given, that is in inverting the linear integral transformation.

[2] The terms gathering and shooting were introduced by Cohen *et al* in the context of explaining the difference in the strategy used by the full radiosity solution and the progressive radiosity algorithms[12].

equations for describing the illumination process. This adjoint system of equations can be seen to provide the mathematical basis necessary to describe all illumination computation methods.

3.1 Deterministic Gathering Methods

3.1.1 Local Illumination Model and Ray Tracing

The simplest approach towards solving the radiance equation is to approximate the integration over the hemisphere by a summation as follows:

$$\begin{aligned} L_o(x,\Theta_x) &= L_e(x,\Theta_x) + \int_\Omega f_r(\Theta_x,\Theta_y)L_o(y,\Theta_y)\cos\theta_i d\omega_i \\ &= L_e(x,\Theta_x) + \sum_{j=1}^{N}\int_{\omega(j)} f_r(\Theta_x,\Theta_y)L_o(y,\Theta_y)\cos\theta_i d\omega_i \qquad (3.1) \end{aligned}$$

where N is the number of discrete small solid angles to which the incoming hemisphere has been divided. A further approximation is to replace the integral term by a simpler expression. One such approximation is given below:

$$L_o(x,\Theta_x) = L_e(x,\Theta_x) + \sum_{j=1}^{N} f_r(j)L_o(j)\cos\theta(j)\Delta\omega(j) \qquad (3.2)$$

where $L_o(j)$ is the average radiance coming from the points visible through the discrete solid angle $\Delta\omega(j)$ around x, and $f_r(j)$ and $\theta(j)$ are respectively the average reflectance and the average incident angle for incidence from j-th discrete set of directions.

Yet another simplifying assumption that can be made is that while the illumination coming from light sources in the hemisphere is significant enough for them to be considered individually, light due to reflection from other surfaces may be summed up to give a uniform illumination called ambient illumination. We therefore now have the following equation:

$$L_o(x,\Theta_x) = L_e(x,\Theta_x) + L_a\rho_a(x) + \sum_{j=1}^{ns} f_r(j)\Delta\omega(j)L_e(j)\cos\theta(j) \qquad (3.3)$$

where L_a is the hypothetical constant radiance (ambient term) from every point in the incoming hemisphere and ρ_a is the direction independent hemispherical reflectance of

point x, ns denotes the number of visible light sources, and
$L_o(j)$, $\Delta\omega(j)$ and $\theta(j)$ are all defined with respect to the j-th light source.

The earliest illumination computation methods use equations similar to Eq. 3.3 and are often said as being based on *local illumination models*[29]. The term local is because the illumination predicted by the equation is due to the direct or local effects of the light sources, with little consideration of global effects such as inter-reflection amongst objects in the environment. Because of their computational simplicity these methods have been used very widely for many years, to produce shaded pictures of 3D objects. The very first equation of this kind is due to Bouknight[8] who gave the expression for the radiance[3] from diffuse surfaces as follows:

$$L_o(x) = \sum_{j=1}^{ns} L_e(j) k_d(x) \cos\theta(j) \tag{3.4}$$

The sum is over ns point light sources,
$L_o(x)$ is the reflected radiance from point x, and because the reflection is assumed to be from a perfectly diffuse surface it is independent of the outgoing direction,
k_d is the diffuse reflection coefficient, which take values from 0 to 1.

Phong[53] subsequently introduced an important improvement to this model for supporting shiny surfaces. Reflection from every surface is assumed to have a diffuse reflection component and an imperfect specular reflection component which is modelled by Phong's reflection model (Eq. 2.5). With this extension Eq. 3.4 takes the following form:

$$L_o(x, \Theta_x) = \sum_j L_e(j) \left[k_{diffuse}(x) \cos\theta(j) + k_{specular}(x) \cos^n \alpha(j)\right] \tag{3.5}$$

where

> $\alpha(j)$ is the angle between the mirror reflection direction of the j-th light and the direction Θ_o,
> $k_{diffuse}$ and $k_{specular}$ are the diffuse and the specular reflection coefficients, which take values from 0 to 1 subject to the condition that $k_{diffuse}$ +

[3] Intensity (I) was the actual term used instead of radiance (L), but with the same meaning. The term intensity is still used by many for radiance. However, as discussed in *Appendix A* they are not the same. In this thesis these two terms will be used in conformity with their exact definitions.

$k_{specular} < 1$,

n is the roughness measure as described in Phong's reflectance model.

Note the directional dependence of the outgoing radiance L_o because of the introduction of non-diffuse reflection behaviour. The L_e term does not appear in Eq. 3.4 and Eq. 3.5 because surfaces are assumed to be non-emitting.

Cook *et al* propose another equation using Phong's reflection model for the $f_{directional}$ term, which is very much like Eq. 3.3 and has the following form[17]:

$$L_o(x, \Theta_x) = \sum_j L_e(j) \left[k_{diffuse}(x) + k_{specular}(x) \cos^n \alpha(j)\right] \cos\theta(j) d\omega(j) \tag{3.6}$$

If we use Eq. 3.4, evaluation of radiance requires the evaluation of the incidence angle θ which in turn requires the evaluation of the surface normal at the visible point. For a polygonal surface, the normal is the same for all points. Therefore when illuminated by light source(s) at infinity a polygonal surface has a uniform value of radiance for all its points, and only a single computation of radiance holds for the entire surface. However, this simplification is not possible for non-planar surfaces. For such surfaces, Gouraud proposed an interpolation scheme, well known as *Gouraud shading*[28]. A polygonal approximation of the curved surface with values for normals at vertex points is first obtained. Using these normals and applying Eq. 3.4, radiance values are computed for all the vertices of this polygonal approximation. Finally for every visible point of a polygonal face of the approximation, radiance is estimated by carrying out bilinear interpolation using the radiance of its vertex points. Phong also proposed a somewhat similar rendering technique using Eq. 3.5 for the visible points of the curved surface[53]. The technique is known as *Phong shading.* Instead of interpolating radiance values, vertex normals are interpolated to arrive at an estimate for the normal at any visible point of the polygonal patch. Eq. 3.5 is then applied to evaluate radiance. This technique is definitely more expensive than Gouraud shading as it makes the evaluation of Eq. 3.5 necessary at every point, but saves on the complex computation necessary for the computation of the actual normal by restricting the expensive computations for obtaining precise normals only to a few points. However surfaces are more smoothly shaded, than in Gouraud shading. Also since specular reflection is properly simulated highlights are obtained.

Rendering a shaded picture is the process of computing shade or colour for the surface points visible through the pixels. The shade due to a point on a surface is proportional to its radiance. Any of the illumination computation methods may be used for computing the radiance. For example one may use equations discussed in this section to compute the radiance at the visible point. In the early rendering techniques visibility computation was carried out mostly by scan-conversion techniques. Extension of these scan conversion methods to compute the visibility of light sources at the point of interest is difficult. So rendering methods based on scan conversion technique for visibility are not generally capable of handling shadows in the environment. Nevertheless, we come across extensions for incorporating shadow computation into these rendering methods[21, 76]. With respect to each light source Crow[21] tagged the portions of the surface patches under shadow. At rendering time these tags are referred to for deciding on whether the contribution of a particular light source is to be added or not. Williams also used a similar concept[76]. Instead of tagging the surfaces, with each light source he associated a raster map of the depth of the points visible to that light source. At rendering time, the distance of the visible point from the light source is compared with the depth stored in the shadow map to see if the point is farther away and hence under shadow.

The ray tracing method, geometrically speaking, is basically a visibility computation method and is ideal for handling shadows from point light sources. In the ray tracing method, determining the surface point visible through a pixel is done by tracing a ray from the viewpoint through a point within the pixel (usually the center of the pixel). The ray is intersected with all the objects in the environment. The nearest of all the intersections is the visible point. A point light source may be hidden to a surface point because of one or more obstructing objects lying between the point and the light source. This information is easily derived by tracing a ray from the point towards the light source and checking if any intersection of the ray with objects lies between the point and the light source.

Ray-object intersections play a major role in the ray-tracing based methods. Fortunately, a ray is a simple 1D linear geometric primitive. Efficient algorithms for computing the intersection of a ray with a large number of object shapes have been devised. That is why a ray tracing method in general does not impose many restrictions on the type of object shapes that it can handle. Since most of the time is spent in computing ray-object intersection, a very large number of extensions are basically acceleration methods for speeding up the ray object intersection computations. A good description of various ray-object intersection and acceleration techniques may be found in [26].

We see in the above that light is gathered accurately only from point light sources and for gathering from everywhere else in the hemisphere a very approximate term called ambient illumination has been used. For a better solution of the radiance equation it is essential that more precise methods of gathering illumination from everywhere else be used. The basic ray tracing method attempts to do this by providing a mechanism to gather information from other dominant directions also. In particular, for calculating illumination from a shiny surface it probes along the mirror reflection direction of the direction of interest by sending a reflected ray. The basic idea behind this extra probe is that light incident on a shiny surface from any direction is mainly reflected along its mirror reflection direction. So if a significant reflection comes from a mirror reflection direction then it is accounted for. Thus for computing this interreflection term, a ray is traced along Θ_y^{-1},(see Fig.3.1) the mirror reflection direction of Θ_x, where y is the visible point when viewed from x along Θ_y^{-1}[75]. The radiance from y, $L_o(y, \Theta_y)$, is added to expression of $L_o(x, \Theta_x)$ in Eq.3.3 after taking into account the losses due to absorption at x and we arrive at the following expression for computing illumination using ray tracing:

$$L_o(x, \Theta_x) = L_e(x, \Theta_x) + \rho_a(x)L_a + \rho_s(x)L_o(y, \Theta_y) + \sum_{j=1}^{ns} f_r(j)\Delta\omega(j)L_e(j)\cos\theta(j) \quad (3.7)$$

where $\rho_s(x)$ is the specular reflectance of the surface to which x belongs.

The difficulty with this equation is that if y does not belong to a light source then $L(y, \Theta_y)$ is also an unknown and hence needs to be evaluated. If this calculation is

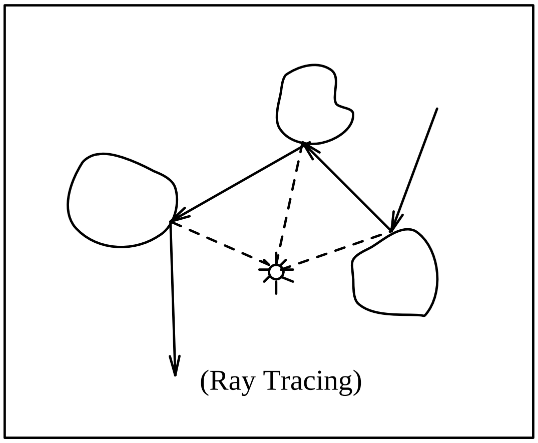

Figure 3.1: Recursive ray tracing.

carried out again using Eq. 3.7, then we say that recursive ray tracing has been used to solve the radiance equation.

Extensions to Participating Medium: The ray-tracing paradigm is simple enough to be extended to participating medium. In a participating medium, radiance in any direction is attenuated by absorption and scattering by the medium, and augmented by scattering of light propagating in other directions. A number of researchers have presented ray-tracing extensions to compute illumination in the presence of participating medium as follows[39, 42, 48]

- The direct and indirect light reaching a reflecting surface is attenuated using Bouguer's equation (Eq. 2.10).
- Radiance along the view ray is the sum of attenuated surface radiance from the nearest surface point and the integrated scattered component coming in from every point along the direction. Light from sources is the only one considered significant enough for scattering computations.

These methods differ primarily in the way in which the integration of the scattering component is carried out.

It must be understood here that very little attempt has been made to accurately model the interaction of the light with the medium. Even with the above simplifications computation times are prohibitively high.

3.1.2 Radiosity

The radiosity method provides a better solution to the radiance equation by accounting for all the integral terms in the summation of Eq. 3.1. The basic radiosity method provides a solution for an environment with all surfaces exhibiting diffuse behaviour[27]. Further improvements are in the form of extensions to support specular surface behaviour and general surface reflectance property. While the feasibility of these extensions to support general surface reflectance property has been demonstrated their application to real situations on a regular basis have yet to be proven[65]. As of today the computational complexity in their implementations do not make them very easy to use.

The basic radiosity formulation is based on the principles of flux transfer in an enclosure. In an environment every object surface can be considered as being completely surrounded by an envelope of surfaces of other solid objects or open areas. This envelope is the enclosure for the surface and it accounts for all directions surrounding the surface. By considering the radiation from the given surface to all parts of the enclosure and the radiation arriving at the surface from all parts of the enclosure, all the radiative contributions are accounted for. Because of this enclosure assumption every solid angle in the incoming or outgoing hemisphere around a point will be covered by a surface. If the environment is discretised to a number of small surface patches then the hemisphere around any surface point can be represented as a sum of solid angles occupied by each visible surface patch on the hemisphere(Fig.3.2). In particular we may represent it as a sum of solid angles due to each of the surface patches, $\sum_{j=1}^{N} \Delta\omega(j)$ where N is the total number of surface patches. A surface completely hidden to a point will have $\Delta\omega(j) = 0$. With the assumption of diffuse reflection behaviour and the further assumption of uniformity of radiance over a patch we get the following simplified

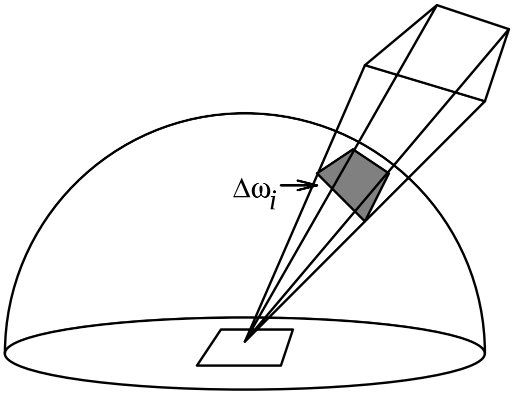

Figure 3.2: Solid angle subtended by a patch over a surface point.

form of Eq. 3.1:

$$\begin{aligned} L_o(i) &= L_e(i) + f_{diffuse}(i) \sum_{j=1}^{N} L_o(j) \int_{\Delta\omega(j)} \cos\theta d\omega \\ &= L_e(i) + f_{diffuse}(i) \sum_{j=1}^{N} L_o(j) \pi F(ij) \end{aligned} \tag{3.8}$$

where $F(ij) = \frac{1}{\pi} \int_{\Delta\omega(j)} cos\theta d\omega$.

We may interpret $F(ij)$ as the fraction of the total outgoing flux from the i-th patch reaching the j-th patch. So $\sum_{j=1}^{N} F(ij) = 1$. $F(ij)$ contains only geometry related terms and hence is termed as *geometric factor*, or *form-factor*. Eq. 3.8 is a simple linear equation, containing geometry dependent and geometry independent terms for obtaining the radiance value at a point on the i-th patch of a diffuse 3D environment. The original equations used in radiosity methods contain radiosity, B, in the equation instead of the outgoing radiance L_o. The term radiosity means the rate of radiant energy outgoing per unit area from a surface. For a diffuse surface radiosity and radiance are related by the expression : $B = \pi L_o$. So we can easily convert the radiance equation (Eq. 3.8) to the radiosity equation by multiplying π on both sides to obtain:

$$\begin{aligned} \pi L_o(i) &= \pi L_e(i) + \pi f_{diffuse}(i) \sum_{j=1}^{N} L_o(j) \pi F(ij) \\ B(i) &= E(i) + \rho_d(i) \sum_{j=1}^{N} B(j) F(ij) \end{aligned} \tag{3.9}$$

where $B(i)$ and $E(i)$ are respectively the total radiosity and the radiosity due to emission from the i-th surface patch, and $\rho_d(i) = \pi f_{diffuse}(i)$ is the diffuse surface reflectance of the i-th patch.

Writing down one equation for each of the surfaces in the environment results in a set of linear equations which can be solved for obtaining the equilibrium radiosity values.

If the environment consists of N patches, then we get a system of N linear equations with N unknowns of the form:

$$\begin{bmatrix} 1-\rho_d(1)F(11) & \dots & -\rho_d(1)F(1i) & \dots & -\rho_d(1)F(1N) \\ . & \dots & . & \dots & . \\ . & \dots & . & \dots & . \\ -\rho_d(i)F(i1) & \dots & 1-\rho_d(i)F(ii) & \dots & -\rho_d(i)F(iN) \\ . & \dots & . & \dots & . \\ . & \dots & . & \dots & . \\ -\rho_d(N)F(N1) & \dots & -\rho_d(N)F(Ni) & \dots & 1-\rho_d(N)F(NN) \end{bmatrix} \begin{bmatrix} B(1) \\ . \\ . \\ B(i) \\ . \\ . \\ B(N) \end{bmatrix} = \begin{bmatrix} E(1) \\ . \\ . \\ E(i) \\ . \\ . \\ E(N) \end{bmatrix}$$

or in short $A\,B = E$.

This set of equations has the unique characteristic of being diagonally dominant and hence is amenable to efficient solution by iterative methods such as Gauss-Seidel's. The major problem in using this method is the setting up of the system of equations. This requires the calculation of all the $F(ij)$ values.

The simplification of Eq. 3.1 to a system of linear set of equations with geometry dependent and geometry independent terms was possible only due to the assumption of the diffuse reflecting nature of the surface patches. Such a simplification cannot be made if we assume a more general form for the surface *brdf* as it will no more be possible to take out the f_r term from inside the integration. Further, because of the diffuse nature there existed a simple relation between radiosity and radiance *viz* $L = B/\pi$, and hence a single radiosity value was good enough to represent the outgoing radiance in any direction. However this does not hold for surface patches with general reflectance behaviour because the complex *brdf* gives rise to complex radiance distribution over the surface patches. Thus it is generally much more difficult to compute the global illumination for general environments using radiosity methods. We shall briefly discuss below a few of the computational strategies that have been suggested for extending the radiosity method to environments with non-diffuse surfaces.

Extensions to Non-Diffuse environment

(i) **Two-Pass Solutions :** Two-pass solutions[11, 66, 71] are based on the assumption that a majority of the surfaces in an environment are diffuse and only a few of the surfaces are non-diffuse. For a non-diffuse surface the radiance distribution is direction dependent. Hence a single radiosity value is not an adequate representation. The strategy adopted is not to associate any radiosity values with the non-diffuse surfaces in the initial pass but to only use them for proper light propagation. That is, use the non-diffuse surfaces to provide indirect light transport paths between diffuse surfaces. As and when needed radiance values for specific directions are computed for non-diffuse surfaces from the surrounding diffuse patches. With this strategy it is possible to formulate a set of linear radiosity equations for the diffuse surfaces of the environment. However the radiosity equations must now include geometry terms, called as *extended form-factors.* The extended form-factor is the fraction of total outgoing flux from the i-th surface patch reaching j-th surface patch directly and indirectly due to one or multiple reflections of this light by non-diffuse surfaces in its propagation path. Though not computationally simple it is possible to compute these extended form factors if we assume that the non-diffuse surfaces in the environment exhibit perfect specular behaviour.

The solution proceeds in two passes. In the first pass the equilibrium radiosity values for the diffuse surfaces of the environment are computed by solving the linear set of equations corresponding to the diffuse patches. In the second pass the radiance values for the non-diffuse surfaces in specific directions are computed by sampling the *brdf* of the non-diffuse surfaces. As most of the surfaces in the environment are diffuse, the sampled directions mostly lead to diffuse surfaces whose radiance values are already known as a result of the first pass.

(ii) **Direction Discretisation :** This method[36] attempts to solve the directional distribution problem by discretising the hemispherical directions around a non-diffuse patch into a finite number of solid angles, within which uniform brightness is assumed. Surfaces are also divided into small discrete patches. For each discrete direction around the patch, a radiosity like linear equation is formulated. Energy leaving every discrete solid angle is obtained by solving the linear set of equations.

For an accurate solution by this method one has to resort to very fine discretisation of the surfaces and also the directions around the patch. Memory requirements are very high thus making it practically impossible for even moderately complex environments. The discrete representation of directions also gives rise to severe aliasing problems.

Extensions to Participating Medium An environment with a participating medium can be assumed to be consisting of surface and volume elements. Rushmeier *et al*[58] have extended the radiosity method to include surface-volume, volume-surface and volume-volume interaction. Illumination is computed by solving a set of linear equations which in addition to the terms for surface radiosity include terms for volume radiosity.

The enclosing medium is subdivided into a number of volumes each of which can be considered essentially to be homogeneous and to be of uniform brightness, similar to the concept of dividing surfaces into smaller patches of uniform brightness. The total energy flux incident upon the patch/volume can be obtained as the sum of contributions from all patches and from all volumes. An energy balance equation is then written for each patch and each volume. This results in a set of simultaneous equations for the unknown fluxes which can be solved. In addition to the geometric terms $F(ij)$ which account for surface-surface transfer, in this extension, we need terms for surface-volume, volume-surface and volume-volume as well. These latter three terms involve computationally intensive integrations. In [58] an approximate evaluation for these terms has been carried out and the results used to compute a rough estimate for the global illumination in the environment. The environment is restricted to contain diffuse emitting and/or reflecting surfaces, and isotropically scattering and nonemitting volumes.

3.2 Nondeterministic Gathering Methods

As mentioned earlier, Monte Carlo quadrature and Random Walk are the two main non-deterministic methods that have been used for gathering illumination from the incoming hemisphere. These methods are called nondeterministic or probabilistic because repeated application of a solution method to the same problem is not guaranteed to give identical results. The methods include steps that depend not only on the input but also on results of some random events.

3.2.1 Monte Carlo Solution of Radiance Equation

The main principle behind a Monte Carlo quadrature for computing the integral $\int F(x)dx$ is as follows[40]:

1. Rewrite *F(x)* as a product $f_1(x)f_2(x)$ such that $\int f_1(x)dx = 1$, *i.e.* $f_1(x)$ is a *pdf*.
2. Sample f_1 for a x_i.
3. For each such sample x_i evaluate $f_2(x_i)$.
4. Carry out the steps (2) and (3) for a number of times, say n. The average, $\frac{1}{n}\sum_{i=1}^{n} f_2(x_i)$, is the estimate of the integral. Larger the n more accurate is the estimate.

The principles of *pdf* sampling are discussed briefly later in Chapter 4.

Using the Monte Carlo quadrature techniques, estimates for the integral part of the radiance equation can be arrived at by simply averaging the radiance from a number of sampled directions from the incoming hemisphere. However, there are a number of problems in this approach. Illumination from the hemispherical directions is due to emission and reflection. Contribution due to reflection is not known and can only be obtained by a similar integration of the hemispherical contributions at the point that is visible along the sampled direction. Along any sample direction only the contribution due to emission is known. The light sources are often localised and contribution from any point on the source is almost always significantly more as compared to the contribution from a reflector. Hence, it is preferable to separate out the hemispherical contribution due to emitters and reflectors in the radiance equation and to use different sampling techniques when evaluating the contributions from the light sources and contributions due to inter-reflection.

Cook *et al*[15, 16] have presented a Monte Carlo solution by using such a breakup and choosing different *pdf*s for different sampling directions for the Monte Carlo evaluation of each term. The modified equation used by them is given below:

$$\begin{aligned} L_o(x,\Theta_x) &= L_e(x,\Theta_x) + L_r(x,\Theta_x) \\ L_r(x,\Theta_x) &= \int_\Omega f_r(x,\Theta_x,\Theta_i) L_e(y,\Theta_i)\cos\theta_i d\omega_i \\ &\quad + \int_\Omega f_r(x,\Theta_x,\Theta_i) L_r(y,\Theta_i)\cos\theta_i d\omega_i \end{aligned} \tag{3.10}$$

where $L_r(x,\Theta_x)$ is the radiance due to reflection only.

The first integral term of Eq. 3.10 accounts for the contribution due to sources and the second term accounts for inter-reflection. The source term is estimated by sampling the light source surfaces and the second term, the inter-reflection term is estimated by sampling the surface *brdf*. This technique is widely known as distributed or distribution[4] ray tracing. Radiance, emitted or reflected, from a sampled direction is arrived at by tracing a ray in that direction, finding the nearest surface point along that direction and computing the appropriate radiance at that point. Thus the *distribution* ray tracing method is a modified recursive ray tracing method, where:

(i) For gathering the contribution from each light source, illumination rays are not traced towards a single light direction, but are distributed according to the emission distribution function of the light source.

(ii) Again for the inter-reflection component, reflected rays are not traced in a single mirror direction but are distributed in the incoming hemisphere according to the bidirectional reflectance distribution function of the surface point.

Distribution ray tracing results in a very accurate solution to the radiance equation *albeit*, at a very high cost due to the excessively large number of rays that need to be

[4] Originally the technique was known as *distributed* ray tracing. However to avoid any confusion with the distributed term normally used for distributed computations in parallel processing, many researchers today including the authors, prefer to refer to it as *distribution* ray tracing[63].

traced. A number of attempts have been made to increase the efficiency of distribution ray tracing. One such attempt is the caching mechanism proposed by Ward *et al*[73] for diffuse environments. In a diffuse environment the *brdf* over a point is uniform in all the hemispherical directions. So the inter-reflection integration component of Eq. 3.7 can be written as a product of the surface reflectance and the incident hemispherical irradiance given below:

$$\int_{\Omega} f_r(x,\Theta_x,\Theta_y)L_r(y,\Theta_y)\cos\theta d\omega = f_{diffuse}\times\int_{\Omega} L_r(y,\Theta_o)\cos\theta d\omega = f_{diffuse}\times irradiance \tag{3.11}$$

So computation of the inter-reflection component at any point requires the evaluation of the incident hemispherical irradiance by distribution ray tracing. A cache is used to store previously computed irradiance values at various points in the environment. When calculating radiance at any point the stored irradiance of the nearby cached point(s) is used. Wherever possible, pixel to pixel illumination coherence is used to estimate the irradiance at any point from the stored irradiance.

3.2.2 Random Walk Solution of Radiance Equation

The radiance equation is an integral equation of the second kind. Estimates of the solutions of such equations can also be obtained by *random walk* methods. A random walk is basically a sequence of steps. Each step is a random sample of the probability distribution function defined over its previous step. Given an integral equation of the second kind, say $\phi(s) = f(s) + \int K(s,t)\phi(t)dt$, if $\int K(s,t)dt = 1$, then $K(s,t)$ at s can be used as a *pdf* . Given a starting point s, the random walk can proceed by sampling the *pdf* to arrive at a random t, and at the point t sample its associated *pdf*, $K(t,u)$ to arrive at a random point u and so on. Based on this we can provide an estimate for $\phi(s)$ as follows:

$$\begin{aligned}\phi(s) &= f(s) + \int K(s,t)\phi(t)dt = f(s) + f(t) + \int K(t,u)\phi(u)du \\ &= f(s) + f(t) + f(u) + \ldots\end{aligned}$$

If $\int K(s,t)dt < 1$ then also one can use the same method by introducing an additional event of absorption into the *pdf*. That means at every step either a next step is chosen according to the probability $K(s,t)$ or no next step is chosen according to the probability $1 - \int K(s,t)dt$. The random walk is bound to terminate and hence the sequence of steps is always finite. The sum of these finite number of f terms then provides an estimate of $\phi(s)$. Applying this technique to the radiance equation where $\int_\Omega f_r(x, \Theta_x, \Theta_i) \cos\theta_i d\omega_i < 1$ gives us the estimate of the radiance as a sum of a finite number of emission radiance values as follows:

$$L(x, \Theta_x) = L_e(x, \Theta_x) + L_e(x', \Theta_{x'}) + L_e(x'', \Theta_{x''}) + \ldots \tag{3.12}$$

where $\Theta_{x'}$ is chosen by sampling *pdf* at x and x' is the surface position visible to x along $\Theta_{x'}$, $\Theta_{x''}$ is chosen by sampling *pdf* at x' and x'' is the surface position visible to x' along $\Theta_{x''}$, and so on. The average of such estimates computed over a large number of paths is used to obtain a more accurate estimate of the radiance value.

Kajiya proposed such a solution to the *rendering equation* (Eq. 2.9), a variant of the radiance equation, and called this type of solution strategy as *path tracing*. The random walk traces a path $x, x', x'', \ldots$ by starting the walk from x to x' where x' is chosen by shooting a ray at a chosen angle (see Fig.2.10) and finding the closest point of interaction and the estimated solution is given by:

$$I(x,x') = I_e(x,x') + \rho(x,x',x'')I_e(x',x'') + \rho(x,x',x'')\rho(x',x'',x''')I_e(x'',x''') + \ldots \tag{3.13}$$

$I(x,x')$, $I_e(x,x')$ are transport intensity terms, and $\rho(x,x',x'')$ denotes a three point reflectance value.

Path tracing differs from distribution ray tracing. In path tracing a single ray emerges from each point where as in distribution ray tracing a large number of rays emerge from a single point. However, the requirement of tracing a large number of paths to get a reasonable estimate of the solution makes it as expensive as distribution ray tracing. A slightly different implementation of the random walk may be carried out by using Eq. 3.10 as the base equation. This is as follows:

1. estimate the source contribution by distribution sampling,

2. estimate the inter-reflection by the random walk.

For this implementation the estimate of $L(x, \Theta_x)$ takes the following form:

$$\begin{aligned}
L(x, \Theta_x) = L_e(x, \Theta_x) \quad &+ \quad \int_\Omega f_r(x, \Theta_x, \Theta_y) L_e(y, \Theta_y) \cos\theta_y d\omega_y \\
&+ \quad \int_\Omega f_r(y, \Theta_y, \Theta_z) L_e(z, \Theta_z) \cos\theta_z d\omega_z \\
&+ \quad \ldots
\end{aligned}$$

This implies that at every step of the random walk we have to compute the hemispherical integration to account for direct contributions. These integrations may be carried out by approximations as used in the local illumination model or may be carried out by multiple sample rays towards each light source as in distribution ray tracing. Though path tracing in general cannot be considered as being very efficient, Kajiya[39] suggests careful use of various variance reduction techniques such as hierarchical and nonuniform sampling to make path tracing an efficient and acceptable alternative for accurate illumination computations.

3.3 Deterministic Shooting Methods

As we said earlier the shooting strategy is a direct simulation of the physical process of light propagation. Sources are primarily responsible for illumination of the environment. If the need is to compute radiance values at a large number of points and not just a selected set, like say points on visible surfaces, then the shooting strategy appears to be natural. In one of the early attempts two pass ray tracing was proposed. The first pass was essentially shooting light from one light source to points arranged in a 3D grid fashion covering the entire volume of the environment[10]. The second pass then was the normal gathering by ray tracing with additional rays for gathering from the 3D grid points. However, the very first proper strategy of shooting light and simulating light propagation for illumination computation was proposed only in 1988[12]. The method widely known as progressive radiosity was proposed primarily as an extension to standard radiosity method to make it computationally efficient. In progressive radiosity the argument in favour of the shooting strategy is stated rather indirectly as follows:

> Radiosity computation of any patch requires the gathering of radiosity from every other patch in the environment. However, only a few of these gathered values are significant enough to contribute towards the brightness of the patch of interest. The significant contributions are mainly due to light sources and bright reflector surfaces (often highly reflecting surfaces which receive light directly from the light sources). So a method which can consider only those significant patches and ignore the rest is likely to be more efficient. This makes finding the set of major contributors important. Light sources are undoubtedly part of the set of such contributors. The other contributors are the ones receiving maximum emitted light directly or indirectly and hence can be found by shooting the light from the source(s) and keeping track of the quantity of light reaching every other surface patch of the environment.

The distinguishing feature of this method is that while one surface is shooting light the outgoing flux of all other surfaces are simultaneously updated. For shooting purposes the surfaces are processed in sorted order according to their flux contribution to the environment. The sorted list of surfaces initially contains only the emitters. As the shooting progresses, the receivers with acquired flux are added into the list. From the radiosity equation, Eq. 3.9, the amount of light received by the i-th patch after a single shooting operation from a bright patch, say j, is given by

$$\Delta B(i) = \rho_d(i) F(ij) B(j)$$

where $\Delta B(i)$ is the amount of radiosity reflected by the i-th surface patch due to the light reaching from the j-th patch, and F is the geometric factor. By progressively shooting light from all the light sources and bright surfaces of the environment the radiosity accumulated at each of the surface patches in the environment approaches the actual equilibrium radiosity value. This is easily shown by the derivations below:

$$B(i) \quad = \quad E(i) + \rho_d(i) \sum_j F(ij) B(j)$$

$$\begin{aligned}
&= E(i) + \rho_d(i)\sum_j F(ij)\left(E(j) + \rho_d(j)\sum_k F(jk)B(k)\right)\\
&= E(i) + \rho_d(i)\sum_j F(ij)E(j) + \rho_d(i)\sum_j \left(F(ij)\rho_d(j)\sum_k F(jk)B(k)\right)\\
&= E(i) + \rho_d(i)\sum_j F(ij)E(j) + \rho_d(i)\sum_j \left(F(ij)\rho_d(j)\sum_k F(jk)E(k)\right) + \ldots
\end{aligned}$$

Extensions to Nondiffuse Environment The shooting strategy can also be used for two pass methods and the directional discretisation method discussed in the earlier section. A recent extension[65] that has been proposed for dealing with non-diffuse surface reflectance behaviour is the use of spherical harmonics for the representation of the directional variation in the outgoing radiance of a point on a non-diffuse surface. It uses a continuous representation of the radiance distribution around a point, instead of resorting only to discrete approximation.

Brief Note on Spherical Harmonics: Spherical harmonics are the elements of an infinite set of orthonormal functions[55], $Y_{l,m}(\theta, \phi)$ of two variables θ, ϕ defined over the sphere, where $0 \leq l \leq \infty$ and ,$-l \leq m \leq +l$ such that for $m \geq 0$

$$Y_{l,m}(\theta,\phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)\exp^{im\phi}$$

and for $m < 0$

$$Y_{l,m}(\theta,\phi) = (-1)^{|m|}Y^*_{l,|m|}(\theta,\phi)$$

where P_l^m are associated Legendre polynomials. Given an arbitrary function in θ, ϕ it is possible to represent it as a linear combination of these spherical harmonics i.e.

$$\begin{aligned}
f(\theta,\phi) &= \sum_{l=0}^{\infty}\sum_{m=-l}^{+l} C_{l,m}Y_{l,m}(\theta,\phi) \approx \sum_{l=0}^{N}\sum_{m=-l}^{+l} C_{l,m}Y_{l,m}(\theta,\phi)\\
C_{l,m} &= \int_0^{2\pi}\int_0^{\pi} f(\theta,\phi)Y_{l,m}d\omega
\end{aligned}$$

The number of terms N depends on the type of function f and degree of approximation. Thus the function $f(\theta, \phi)$ can be represented as a set $\{C_{l,m} \mid 0 \leq l \leq N\ ,\ -l \leq m \leq +l\}$. Using this technique the f_r of a surface can be approximated as a set of coefficients.

However f_r is a function of $\theta, \phi, \theta_i, \phi_i$ where θ_i, ϕ_i represent the incoming directions and θ, ϕ represent the outgoing directions. As one is interested in the outgoing distribution of the reflected radiance, it is more appropriate to represent $f_r(\theta, \phi, \theta_i, \phi_i)$ as a set of $\{C_{l,m}(\theta_i, \phi_i) \mid 0 \leq l \leq N \ , \ -l \leq m \leq +l\}$. For surfaces with isotropic reflections *i.e.*, those surfaces with f_r independent of the circumferential angle of the incoming direction, the coefficients will only be dependent on θ_i and hence will be represented as a set of $\{C_{l,m}(\theta_i) \mid 0 \leq l \leq N \ , \ -l \leq m \leq +l\}$.

3.4 Remarks

Illumination computation has been one of the most extensively researched subjects in the field of computer graphics. The total amount of published work is enormously large and it would not have been feasible or beneficial to attempt to accommodate all of these in our review. However the treatment of these methods as being algorithmic solutions to the radiance equation, and the categorisation of strategies into gathering vs shooting in one dimension and deterministic vs nondeterministic along another is the first of its kind and has ensured that the more significant contributions have all been adequately covered.

There are three important observations which must be put on record:

(1) In the early days of illumination computation, simulation of light propagation was tightly coupled with rendering and in many places indistinguishable. As methods evolved to greater degrees of sophistication the need for decoupling these two aspects of the problem was clear. With the introduction to the graphics community of the radiosity method and of the rendering equation this decoupling was complete. The radiosity method showed that for completely diffuse emission/reflection behaviour, illumination computations are independent of viewing parameters and can be performed on a number of wavelength bands. If the lighting conditions do not change the computed radiosities also do not change and hence any number of views can be generated once the radiosities of all the surfaces are computed and known.

(2) By taking into consideration all light interacting elements of the complete diffuse environment the radiosity method yields accurate global illumination effects with greater computational efficiency than other non-deterministic gathering methods like distribution ray tracing and path tracing. Colour bleeding across surfaces, area light sources, variable shading within shadow envelopes, penumbra effects along shadow boundaries etc. are some of the effects which can be properly captured resulting in highly realistic imagery when rendered. However as this computational efficiency is derived directly from the deterministic approach, it must be recognised that the solution is only as accurate as the discretisation of the environment. In particular, when we consider extensions to the radiosity based solution for general environments with non-diffuse behaviour as well, the discretisation of the environment has to be extended to accurately accommodate directional dependence of radiosity. The problem of automatically carrying out accurate discretisation of an environment is being separately addressed by many but the fact remains that in spite of all the extensions and the fantastic quality of the sample scenes rendered using these methods, the basic deterministic approach is still a restrictive solution to the integral equations characterising the simulation of light propagation globally in an environment.

(3) The strategy of shooting light starting from light sources can be seen as very natural for simulating the propagation of light in an environment. The accuracy of the computed illumination is continuously refined as the simulation progresses. Depending on the need in any situation, the computation can be terminated resulting in dramatic reduction in computation times for reasonably accurate solutions. However, the requirements of accurate discretisation remain and hence the difficulties to deal with more general environments continue.

It is important to emphasise here that the gathering strategy based methods can all be treated as methods providing solutions to the basic radiance equation required to simulate the light propagation process. The nondeterministic gathering methods are general solutions to the radiance equations and hence in principle can deal with all kinds of general environments in the same manner. Also when seen as solutions to the governing radiance equation it is possible to deviate from the requirement of physical simulation and derive more efficient solution techniques where the intermediate steps no longer have physical equivalents. Such efficiency improvements are already in extensive use in other disciplines where similar equations are required for simulations. The non-deterministic solution strategies are usually termed as Monte Carlo methods and the efficiency improvement techniques usually go by the name of variance reduction techniques.

The shooting strategy on the other hand cannot be seen directly as a solution to the radiance equation. It is true that it physically simulates the light propagation process and it is also true that it is intuitively clear that in the limit as the simulation progresses the resulting radiance values accumulated at different points of the environment do approach the physical equilibrium values. However, so far there has been hardly any serious attempt to consider the non-deterministic shooting strategy as an alternative for view independent illumination computations in a general environment. In this thesis we shall primarily explore this strategy and examine its computational feasibility. Later in this thesis we shall also derive the governing equation which has been termed as the potential equation that is required for simulating light propagation using the shooting strategy. Naturally, variance reduction techniques for efficiency improvements are also explored. The potential equation and the radiance equation form an adjoint system of equations which together characterise all the illumination computation methods.

Chapter 4

Particle Tracing: A Nondeterministic Shooting Method

The particle model of light is a natural candidate for use when simulating the propagation of light using the shooting strategy. In this model packets of energy, particles, are shot out or emitted in different directions from different positions on the surfaces of the light sources. Particles move in a straight path and hit other objects, hence forward termed receivers. At the surface of the receiver a particle is either absorbed, thereby loosing all its energy to the receiver or is reflected thereby changing its direction. The new direction acquired by the particle is determined by the direction from which the particle hits the receiver and the *brdf* of the receiver surface. The particle continues its flight until it is eventually 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 particles leave the surfaces in an environment. This rate (flux) determines the brightness of the objects. The distribution of the wavelengths among the outgoing particles from the surface of an object determine the colour of the object. Using this model, illumination is determined by finding the particle flux per unit area at different wavelengths for each surface of each object in the environment.

In complex environments, as already mentioned, where the radiance equation gets extremely complex to solve, this particle flux can be estimated by carrying out a time-independent simulation of the behaviour of a sufficiently large sample of particles and keeping track of their histories. As has been shown in other disciplines such as Neutron Transport [43] and Heat Transfer [35], such simulations are best carried out using Monte Carlo methods. We shall discuss here the use of these methods for simulating the distribution of light. We term the process as *particle tracing.*

Monte Carlo methods are based on the random sampling process. In all Monte Carlo calculations it is necessary to draw samples from some parent population through sampling procedures governed by specified probability laws. In particle tracing the simulation is carried out using a finite number of particles. Each particle originates from a light source. It therefore has to be first assigned an emitter surface, then a position on the emitter surface, then a wavelength and finally a direction of emission. All assignments are done making random choices which over a sufficiently large sample would match the emission behaviour of the emitters in the environment. If we take the example of assigning the emitter surface to the particle, the random choice should be such that the number of particles assigned to the light sources are in proportion to their emissive power. In other words it can be said that the probability of a particle 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. A uniform way of carrying out this sampling is to associate all the possible outcomes with a proability value. In other words associate with each each behaviour a probability distribution function. Then random sample this probability distribution function. Usually the probability function is either a cumulative function, called cumulative distribution function(*cdf*), $F(x)$, or a density function called a probability density function(*pdf*), $f(x)$. Both of these functions are related to each other by the equation $f(x) = \frac{dF(x)}{dx}$. If we assume that an event is the outcome of some stochastic experiment then $F(x)$ is the probability of the event taking any value less than x and $f(x)dx$ is the probability of the event taking any value in the range dx around x. $F(x)$ is a monotonically increasing function of x and $0 \leq F \leq 1$. If the outcome of the events is bounded, *i.e.* $a < x < b$ then the following equations are satisfied:

$$\int_a^x f(x)dx = F(x) \geq 0, \quad F(a) = 0, \quad \int_a^b f(x)dx = F(b) = 1$$

For example in simulating the emission of light particles from a diffusely emitting surface the cone angle, θ, of the emission direction is governed by the *pdf*, $2 \sin \theta \cos \theta$, with *cdf* given by $\sin^2 \theta$.

For each emitted particle, the nearest receiver in the environment along its associated direction is then determined. At the receiver surface the particle is absorbed or reflected. The probability of either of these happening is determined by the surface reflectance. If the particle gets reflected then the particle is registered in the outgoing flux of the receiver surface. The particle is then assigned a new direction and once again the nearest receiver surface along its new path is determined. This process is termed *particle tracing*. For every particle generated this particle tracing is continued till it is absorbed.

On completion of tracing of all the particles we get a simulated particle flux for each of the surfaces in the environment. This simulated flux is an estimate of the actual flux in the real environment. From the *law of large numbers*[1] we know that larger the number of samples better is the agreement of the estimator with the actual value. It is not clear how large is large. Certainly the number of particles beyond which appreciable improvements in the simulation results are not obtained would depend on the actual configuration and complexity of the 3D environment. There does not seem any simple analytical method for determining this number either. An acceptable solution would be one in which this number is determined as the simulation progresses based on the actual changes that occur in the simulation results for any given 3D environment. And such a strategy is discussed in greater detail later in this thesis. It is important however to get some idea of these numbers in order to be assured that the simulation strategy is practical for reasonably complex environments with the kind of computer power available today. And so for the present we shall just get a feel for this number based on extensive experiments that have been carried out with an actual implementation of

[1] **Law of Large Numbers :** In a stochastic experiment with probability density function $f(x)$, the expected outcome, μ, of the experiment is defined as $\int_a^b x f(x) dx$. If, in this stochastic experiment, $x_1, x_2, \ldots, x_n$ are all randomly drawn events then the law of large numbers states that the probability of the the arithmetic mean of these random variables becoming equal to μ tends to 1, *i.e.* $P(\sum x_i/n = \mu) \rightarrow 1$, as $n \rightarrow \infty$.

this algorithm.

Another issue that needs to be addressed is as follows:

How good or accurate are the simulation results?

Carrying out the simulation for a large number of representative 3D environments and then visualising the rendered images is one possible way. However, a more quantitative approach would be to compare results for identical 3D configurations with a more analytically accurate method such as the full radiosity method. A straight forward implementation of the full radiosity method was undertaken and the comparative results obtained for a simple test environment are also presented in this chapter.

4.1 Sampling Techniques

As discussed earlier, a stochastic or random behaviour can be characterised by a mathematical function, called probability density function, f or cumulative distribution function, F. If such behaviour has to be simulated then one requires a method of generating events such that

- each event is independent of the other,
- each event is representative of the behaviour simulated, and
- a large number of such events approximate the total behaviour.

This is termed as sampling. We give below the description of the sampling methods used in our simulation. For the details of various sampling methods, the interested reader is referred to standard Monte Carlo texts[31, 40, 43, 56, 68]. All these methods depend mainly on a uniform random number generator, one which generates random values in the range 0 to 1 with uniform probability.

4.1.1 Sampling Discrete Probability Distribution

If the outcome of a stochastic experiment can take only a finite number of values, say L, then the probability distribution of the outcomes is said to be discrete. The probability of an outcome taking a discrete value l is given by f_l where $0 < f_l \leq 1$ and $\sum_{l=1}^{L} f_l = 1$. For random sampling such distributions the following method can be used.

It is possible to take the interval (0,1) and exhaust it by dividing it into L segments each of which has a length equal to some f_l. If ξ is the uniform random variable in the range (0,1) then the interval into which ξ falls determines the identity of the event. So the procedure is as follows:

> Generate a uniform random variable, ξ.
>
> Find the smallest $m \leq L$ for which $\sum_1^m f_l \; > \; \xi$, *i.e.*
>
> $$\sum_1^{m-1} f_l < \xi \leq \sum_1^m f_l$$
>
> Then m is the necessary sample.

Whenever $0 < \xi \leq f_1$, event 1 takes place, if $f_1 < \xi \leq f_1 + f_2$ then event 2 takes place and so on. For example the interaction of the light particle on a opaque surface may be thought of as a discrete event with two possible outcomes, either absorption or reflection. If ρ is the directional hemispherical reflectance, then we can assume ρ to be the probability of reflection and $(1 - \rho)$ the probability of absorption. So the discrete *pdf* is $\{\rho, (1 - \rho)\}$. By the above method, if $0 \leq \xi \leq \rho$ then the sampled event is reflection otherwise the event is absorption.

4.1.2 Sampling Continuous Distribution

Transformations of Random Variables : As in the case of the discrete distribution sampling, the basic idea behind this method is to provide a mechanism to transform a uniform random variable into a random sample in the required distribution. The sampling method is as follows:

> Generate a uniform random value, ξ in the range (0,1).
>
> Find X such that $\int_{-\infty}^{X} f(x)dx = F(X) = \xi$, *i.e.* $X = F^{-1}\xi$.

In fact this method can be seen as the direct extension of the discrete distribution sampling technique in which the summation has been substituted by integration.

This method is applicable where an explicit analytical form of the distribution function exists and it is possible to derive the inverse of the cumulative distribution function. For example: in the assigning of a direction to the emitted particle from a diffuse emitter the *pdf* for θ in the range 0 to $\pi/2$ is $\sin 2\theta$ and so the *cdf* is $\sin^2\theta$. So by the sampling method given above

$$\sin^2\theta = \xi \quad \textit{or} \quad \theta = \sin^{-1}\sqrt{\xi}$$

Thus the angle θ for the sampled emission direction from a diffuse emitter is simply the sine inverse of the square root of a uniform random number in the range 0 to 1.

Rejection Technique : This technique is computationally expensive and is to be used as the last resort. It can be applied to any distribution function. The general idea behind the method is as follows:

> Propose a trial value for the event.
> Subject this trial value to one or more tests. On the basis of the outcome of the test either accept or reject the proposed value as the sample.
> If the proposed value is to be rejected, then repeat the process till a value gets accepted.

The commonly used method for rejection sampling a density function $f(x)$, bounded in the interval (a, b), is as follows:

> Generate a pair of uniform random numbers (ξ_1, ξ_2) in the range 0 to 1.
> If $[\xi_2 . \sup f(x)] < f(a + \xi_1(b - a))$
> then accept $a + \xi_1(b - a)$ as a sample from $f(x)$.
> If not reject the random pair and repeat the process.

In order to use the rejection method it is necessary to find $f_1(x) = \sup f(x)$, the lowest upper bound for $f(x)$, or at least an upper bound for $f(x)$. However, if only a weak upper bound is found, the efficiency of the rejection method suffers considerably. Also it is necessary that the function $f(x)$ be bounded. The rejection technique suffers from the defect that not all of the random pairs (ξ_1, ξ_2) result in a sample point drawn from $f(x)$. The efficiency of such a method is the ratio of the area under the curve $f_1(x)$ to the area of the enclosing rectangle. As the area under the original function $f(x)$ is 1, the ratio and hence the efficiency is at best $\frac{1}{(b-a)f_1(x)}$.

4.2 Particle Tracing : The Monte Carlo Simulation Method

4.2.1 The Algorithm

To start with we shall make the following assumptions:

- The medium in the environment does not interact with the particles[2].
- The objects are all opaque and are described by their bounding surfaces and associated optical properties such as emission flux distribution, emission spectrum and *brdf*.

Then the algorithm is as follows:

Decide on the number of particles to be traced and for each particle carry out steps (1) to (3) below:

1. Choose
   (a) Wavelength of the particle by sampling the emission spectrum.
   (b) Position of the particle on the emitter surface by sampling positional emission strength distribution.
   (c) Initial direction of the path of the particle by sampling directional strength distribution.
2. Update the outgoing particle flux at the emitter surface.

[2] In the next chapter we shall discuss the extensions of this algorithm to environments where this simplifying assumption is not necessary.

3. Repeat steps (a) to (c) below until the particle is absorbed.

   (a) Find the nearest surface along the particle path.

   (b) Choose the type of interaction i.e. *absorption* or *reflection* by sampling the discrete interaction distribution function $\{\rho, 1-\rho\}$.

   (c) If the interaction is *reflection*

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

      ii. Assign a reflected direction to the particle by sampling the *brdf*.

Section 4.3 discusses in detail the position sampling formulae and methods for different emitter surface types and direction sampling for emitted particles based on the directional emission flux distribution of the source. The path of a particle is assumed straight. So finding the nearest surface hit by the particle (step 3.a) is done by carrying out ray-surface intersections. This has been researched extensively in computer graphics [26]. Methods for sampling different *bdrf*s (step 3.c.ii) are discussed in Section 4.4.2.

Computing Flux Density

To start with we shall record the outgoing flux at each surface patch 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 that subpatch. The relation is as follows:

$$Outgoing\ Flux = Number\ of\ Particles\ leaving\ the\ patch\ *\ Particle\ Strength$$

where $Particle\ Strength = \frac{Total\ Source\ Strength}{Total\ Number\ of\ Simulation\ Particles}$.
From this the outgoing flux density is computed as: $Flux\ Density = \frac{Total\ outgoing\ Flux}{Patch\ Area}$

In the process of simulation a region of the surface under shadow will have very few particles while another region of the same surface directly facing a light source will have a very large number of particles reflected from that region. Thus the positional distribution of the reflected particles directly gives us the variation of brightness over

Figure 4.1: Particle distribution on a suface of an example scene.

the surface. This is clearly illustrated by the example scene shown in Fig.4.1(a) along with Fig.4.1(b) showing the scatter plot of the particles reflected from a surface on the scene. Therefore if in the process of simulation we capture the positional distribution of particles then these simulation results can be interpreted not only to determine radiance in a region but also to determine how its gradient varies over the surface.

A naive approach to the capture of this distribution would be to record for every reflected particle the position on the surface. 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 particle 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 particle 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.

The other approach will be that which is followed in the radiosity methods. We have a predefined mesh structure associated with the receiver surface. The particle flux is assumed to be uniform over a single mesh element and hence a simple count of the number of particles emitted/reflected is sufficient. However in such a case, 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. The automatic discretisation problem is being researched extensively[5, 9, 47] 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 and reflectance 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. For the present let us assume that the particle distribution is accumulated over a rectangular mesh (in *uv* space) imposed over each surface.

4.2.2 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 particles or so have been traced, 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 particles have been traced we have illumination information which is not likely to be drastically different from the result derived after a larger number of particles are traced. 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	Adjacent Faces				Opposite Face
1000	0.6175	0.6216	0.6131	0.6143	0.6038
10000	0.6258	0.6278	0.6250	0.6302	0.6110
100000	0.6299	0.6306	0.6278	0.6294	0.6191
1000000	0.6288	0.6288	0.6283	0.6285	0.6140
Radiosity Method	0.6265	0.6265	0.6266	0.6265	0.6093

Table 4.1: Comparison of photon flux densities obtained from Monte Carlo Simulation method and Radiosity method.

particles or so. 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.

4.2.3 Comparison with Radiosity Method

For comparison purposes, we have carried out a simple implementation[50] of Full Radiosity solution using the hemicube method [13] 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. 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 × 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 4.1 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.4.2 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

(a) Radiosity (303 Patches, 4:42 mins).
(b) Monte Carlo with 100,000 Photons (113 Surfaces, 5:09 mins).

Figure 4.2: Sample results from radiosity and particle tracing.

environment contains a total of 113 polygonal surfaces with larger surfaces selectively broken down to a 10 x 10 grid of patches. Fig.4.3 shows the images rendered using the simulation results after the tracing of 1000, 10,000 and 100,000 particles 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[3] 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 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 in geometrical and optical complexity) with greater flexibility, simplicity and speed.

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

(a) 1000 particle (113 Surfaces, 0:03 mins).

(b) 10,000 particle (113 Surfaces, 0:31 mins).

(c) 100,000 particle (113 Surfaces, 5:09 mins).

Figure 4.3: Progressive refinement in particle tracing.

4.3 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 [70]. They are *geometry* (emitter surface shape), *spectral distribution* (emitter strength at different wavelengths) and *luminous radiance 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 radiance distribution influences the relative brightness of the objects positioned around the emitters. So far only spectral distribution is handled reasonably well 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[29]. The effects of extended light source geometry and anisotropic luminous radiance 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.

4.3.1 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[49] 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 [27]. As discussed earlier in Chapter 3, 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 also be easily incorporated.

Sampling of Position for Emission

As stated earlier, during the simulation process all the particles are assigned positions on the emitter surface. If we assume that the emission strength is uniform over the surface of the emitter[4], then the only requirement for using any arbitrary geometry is that we must be able to generate particles uniformly over the surfaces. In other words the particle density (particles/unit area) must be the 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.

A rectangle or parallelogram shaped surface is one of the simplest of surfaces 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:

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

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 distribution of particles 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.

[4] In almost all situations this will be true. Otherwise a single emitter could be suitably treated as an ensemble of smaller emitters with that property.

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 particles. 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.4.4)

i) Sample along the u parameter by assigning a uniform random number to u in the range 0 to 1.

ii) Sample along the v parameter using the principle of *transformation of random variable.* 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.4.4, for the triangle the strip is rectangular with width b and height dh, for the sphere, and for 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 particle coming out from within a strip is:

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

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.

Figure 4.4: Parameter directions for different geometric shapes.

Using these principles sampling a spherical surface for uniform points is done as follows. The spherical co-ordinates of a point on the sphere, $(\theta,\ \phi)$, are used as parameters instead of u and v. The circumferential angle ϕ is sampled uniformly by:

$$\phi = 2\pi R_\phi$$

To satisfy the uniform sample density requirement the probability of a sample appearing in the ring of width $d\theta$ and radius r is its area divided by the unit hemisphere area.
Area of the ring $= 2\pi r = 2\pi \sin\theta\, d\theta$.
Area of the unit hemisphere $= 2\pi$.
The probability $p(\theta)d\theta = \sin\theta\, d\theta$ in the θ range of 0 to $\pi/2$. Using sample transformation principle:

$$\begin{aligned} \int\limits_0^\theta p(\theta)d\theta &= \int\limits_0^\theta \sin\theta d\theta = R_\theta \\ | - \cos\theta|_0^\theta &= R_\theta \\ \cos\theta &= 1 - R_\theta \end{aligned}$$

For the triangle the derivations are as follows:

- 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 $v = 1 - \sqrt{\xi_v}$.

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	$\frac{1}{2}\pi sin\pi v$	$\frac{\cos^{-1}(1-2\xi_v)}{\pi}$	$(R\sin\pi v\cos 2\pi u, R\sin\pi v\sin 2\pi u,$ $R\cos\pi v)$, for a sphere with centre at origin.
Cylinder	1	ξ_v	$(R\cos 2\pi u, R\sin 2\pi u, vH)$ 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 4.2: Position sampling equations.

For each of the shapes supported in our implementation Table 4.2 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 particle density over the entire surface also means uniform particle density over the smaller parts of the surface. The distribution of particles 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 particle will originate.

4.3.2 Spectral Distribution

The emission spectrum of the light source gives the spectral density function of the emitted light. The emission spectrum is the radiance vs wavelength curve and describes the relative proportion of the wavelength packets emitted at any given time. This information is usually associated with a source specification. Given this spectral density function, in the form of emission spectrum, rejection sampling can be used to decide on the wavelength of a packet emitted at random. In the limit, if the source has an equal energy spectrum one simply uses uniform sampling of wavelengths from the 380nm to 770nm.

4.3.3 Luminous Radiance Distribution

Generally in computer graphics emitters are restricted to have uniform radiance 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 [74]. 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 radiance distribution can also be supported quite easily in Monte Carlo simulation. Supporting any complex emission distribution means assigning a direction to each emitted particle in such a way that the distribution of the particle samples match with the emitter's radiance distribution.

A direction is defined by a pair (ϕ, θ) where ϕ is the circumferential angle and θ is the cone angle. Generally it is assumed that the emission around a point is circular symmetric, *i.e.* is independent of the circumferential angle ϕ and hence radiance distribution is specified as a function of the cone angle by goniometric diagrams [70]. 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.

A particular case is *fixed directional emission*. In this distribution the emission is only along one direction and hence in such an emitter the generated particle 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, say, a window pane.

Direction Sampling for Diffuse Emitters : Yet another special case is diffuse emission wherein the emitted radiance is uniform in all directions and total emmissive power per unit area is π times the emission radiance. With ϕ sampled as explained in the above paragraph, the sampling procedure for θ is as follows:

Let the hemisphere in Fig.4.4 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 particle 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.

Area of the differential strip : $2\pi \sin\theta d\theta$

The light reaching C from P= $2\pi L_e dA \cos\theta \sin\theta d\theta$

The total light emitted from $P = \pi L_e dA$. So

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

By the principle of *Transformation of Random Variables*:

$$\xi_\theta = \int_0^\theta p(\theta)d\theta = 2\int_0^\theta \cos\theta \sin\theta d\theta = \sin^2\theta \ or \ sin\theta = \sqrt{\xi_\theta}$$

Thus for the particle 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.

4.4 Illumination of 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.

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.

4.4.1 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 particle's path. As the particle 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 [26]. 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 also be added without too much effort.

4.4.2 Complex Surface Reflectance

The *brdf* of a receiver determines the distribution of reflected radiance around its surface. In the simulation, surface *brdf* is used to choose the reflection direction for a particle. We consider the two idealised reflectance behaviours - diffuse reflectance and mirror reflectance. In diffuse reflectance the reflected radiance distribution is uniform around the surface similar to the diffuse emission process discussed in Section 4.3.3. 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, similar to the direction chosen for a diffusely emitted particle. For mirror reflectance the choice of direction is simpler. 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. For a surface whose *brdf* is given by Phong's specular model the sampled reflection direction is given by the vector which makes angle $(cos^{-1}(1-\xi_1)^{\frac{1}{n+1}}, 2\pi\xi_2)$ with the mirror reflection of the incident vector[61]. In the above formula n is the empirical surface roughness parameter.

The complex reflectance modelled by Ward's anisotropics reflectance model[72] (Eq. 2.6) could be sampled by assigning

$$\delta = \left[\frac{-\log\xi_1}{\cos^2\phi/\alpha_x^2 + \sin^2\phi/\alpha_y^2}\right]$$

$$\phi = \tan^{-1}\left[\frac{\alpha_y}{\alpha_x}\tan(2\pi\xi_2)\right]$$

where (δ, ϕ) 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** directions respectively.

For sampling more complex *brdf*s one may need to use rejection sampling.

4.5 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 radiance of the light emitted or reflected by a surface and reaching the eye along the view direction. As discussed earlier, the simulation provides us with flux densities over the surfaces of the scene. The flux density and radiance are related by the general equation:

$$\Phi = \int_{\Omega} L(\theta,\phi)cos(\theta)d\omega$$

where $L(\theta,\phi)$ is the radiance along the (θ,ϕ) outgoing direction. For a diffuse surface the radiance of reflected light is constant in all directions. With $L(\theta,\phi)$ independent of θ,ϕ the above equation simplifies and we get $L = \frac{\Phi}{\pi}$. However, such simplifications are not possible for other types of surfaces. Computing $L(\theta,\phi)$ requires the knowledge of directional distribution of the computed flux density. This would require the capture of outgoing particles 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 particles for accurately capturing both positional and directional distribution and hence computation and memory overheads will be prohibitive. In order to overcome this problem we too have adopted the two pass strategy in which:

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

- Non-diffuse surfaces do participate in full in the simulation. Particles 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 particle 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[38]. Any of the established techniques of scan conversion, subdivision(dicing) or ray tracing could be used for determining visible parts [25]. However, once it is established that through a particular pixel a non diffuse surface is visible then from that surface, radiance 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 [71] 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 the reflectance distribution sampling for the incoming diffuse radiance from each surrounding direction can be carried out. 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 radiance from such surfaces has to be computed by applying the sampling method recursively[5]. As in all two pass methods 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.

[5] This strategy of estimating the radiance 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.

Photons	*Simple Absorption(SA)*	*Russian Roulette(RR)*	*RR 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 4.3: Relative performances of simulations based on Simple Absorption and Absorption Suppression models.

4.6 A Variation in the Simulation Algorithm

In the algorithm presented in Section 4.2.1, a particle intersecting a surface is absorbed if a uniform random number drawn at that point is greater than ρ. This is based on the assumption that the interaction is a discrete distribution of two events: reflection and absorption with distributions $(\rho, 1-\rho)$ respectively. Hence the method of absorbing the particle is in accordance to the principle of *Discrete Distribution Sampling.* Every time a particle 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 ρ, the probability of a particle undergoing the first reflection is ρ, the second reflection ρ^2, the third reflection ρ^3 and so on. So the average relative brightness contribution made by each particle to the given enclosure is given ideally by the factor:

$$B_{av} \ = \ 1 + \rho + \rho^2 + \rho^3 + \ldots = \frac{1}{1-\rho}$$

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 *variance.*

In column II of Table 4.3 we have summarised the average brightness contribution made by a particle in our test environment of Section 4.2.3, 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 4.3 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 [35, 43] is based on the following idea: On every interaction instead of choosing to absorb the particle its brightness contributing capacity (*weight*) is reduced and the particle is always allowed to continue. The particle starts with a weight of 1 from the source. On every encounter with a surface its weight is reduced by a factor of ρ and this modified fractional weight is contributed to the brightness of the reflecting surface. However, direct use of this method is impractical, as the tracing of a single particle never terminates. Termination of this process after the particle'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[6] may be used to reduce this bias. The technique is as follows:

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

The computational equivalent of the Russian Roulette [43] is to choose a predefined number $\mathcal{N}$ between 2 to 10. Once the weight of the particle reduces below a sufficiently small threshold a uniform random number ξ is generated. The particle is removed from the system only if $\xi > 1/\mathcal{N}$. The particle 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 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 and on each interaction with a surface the particle'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:

$$Outgoing\ Flux = Total\ weight\ of\ the\ Particles\ Leaving\ the\ Patch * Particle\ Strength$$

[6] 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".

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 4.3. 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 4.3). The computational efforts have been measured by the number of ray-surface intersections. These results too show an improvement over the simple absorption method.

Both methods have been incorporated in the implementation. Either one may be chosen for any simulation run. Our experience so far has been that for equivalent computational efforts there is not much qualitative visual difference between the images created using simple particle absorption model and Russian Roulette model. This may perhaps be attributed to the fact that the total number of particles 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 will yield better results for the same computational effort.

4.7 Remarks

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 particles 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 applicable. There are no known methods to estimate such numerical uncertainties either.

With our implementation we have traced around a million particles for the environment shown in Fig.4.5 which has 661 surfaces, small and large[7]. 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 tracing 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. This means that the tracing of one million particles 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 have found that with the spatial subdivision based acceleration technique that has been implemented the average time for *particle tracing* is more or less independent of the geometric complexity of the environmental configuration.

Particle tracing, thus, provides a very simple method for the computation of global illumination in a 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 can be treated with greater flexibility, simplicity and speed.

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

Figure 4.5: A Complex 3D Scene.

Chapter 5

Particle Tracing in Environments with Participating Volumes

As has already been mentioned by us earlier, one of the principal advantages of simulation using the non-deterministic particle tracing technique is its inherent ability to accommodate with comparative ease the behaviour of light in complex environments. In Chapter 4 we have already shown to some extent how complex surface emission and surface reflection are handled. In this chapter we shall discuss its extensions to environments with participating volumes. When the participating volume is the entire 3D enclosure comprising the complete environment with other objects embedded within it then it is often referred to as a participating medium. An environment could also include geometrically localised participating volumes like a small puff of smoke in a large room. As discussed earlier in Chapter 2, light when traveling through a participating volume may either get attenuated or augmented. Attenuation is due to absorption or scattering of the light by the participating volume and augmentation is due to scattering of light into the light path by other participating volumes or by emission of light within the participating volume. This behaviour of light is modelled using light particles as follows:

1. Particles can originate either from emitting surfaces or from emitting volumes.
2. During their flight through a participating volume somewhere within the volume some of the particles are absorbed, some others are scattered in different directions while others continue unhindered.

Precisely what percentage of the particles get absorbed, what percentage get scattered and where along their path this interaction with the volume takes place depends on the optical properties of the volume such as opacity and scattering albedo. 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.

The particle tracing technique of Chapter 4 is extended to deal with participating volumes as follows:

> Particles are now generated originating either at different positions on the emitter surfaces or at different positions in the emitter volume, and as before are assigned different directions of propagation. An additional step is added to account for the interaction of the particle with the intervening volume.

This step is as follows:

> 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 in the volume at the computed position with a probability determined by the absorption/scattering albedo. A scattered particle is assigned a direction for continuing its flight.

But for this step the simulation progresses in the same way as discussed earlier in Chapter 4. Global illumination is then computed as the scattered particle flux in the volume elements and the reflected and transmitted particle flux at the surface elements.

5.1 Interaction in Absorbing and Scattering Medium

In the additional step discussed above a primary requirement is the following:

> "for every particle moving through a participating volume determine the point of interaction, if any, in the volume."

For this we first set up a *pdf* that models the interaction of light with volume as a function of distance travelled and then sample this *pdf*. From Bouguer's law (Eq. 2.10) the radiance of light on traveling a pathlength S inside a participating medium with extinction coefficient *K(u)* reduces by the factor $e^{-\int_0^S K(u)du}$. This factor may be interpreted as the probability of any particle traveling a path length S before it interacts with the volume. Thus the probability of a 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 principles of Random Variable Transformation,

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

where ξ is the uniform random number. For ξ uniformly distributed over the range 0 to 1, $(1-\xi)$=ξ_1 is also uniformly distributed over the same range 0 to 1. So the sampling equation is

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

For a homogeneous medium the *Opacity* is $K.S$ 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 us to evaluate the integral. This is difficult. Howell[64] 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. The exact point of interaction or the path length S is given by

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

Though the 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 traced through a list of *voxel*s very simply by using the 3D-DDA algorithm 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+$\Delta S_{voxel}$
```

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

For tracing the voxels traversed by a particle along its path the 3D-DDA algorithm is used.

5.2 The Simulation Algorithm

The extended Monte Carlo simulation algorithm for dealing with participating medium is given below:
For each particle repeat steps (1) to (6) below:

1. Choose a wavelength for the particle 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) to (c) below until the particle is absorbed.

   (a) Find the nearest surface along the particle path, and find its pathlength i.e. *surface_interaction_pathlength.*

   (b) Find the *volume_interaction_pathlength* by using the computational method discussed at the end of the previous section.

   (c) If *volume_interaction_pathlength* < *surface_interaction_pathlength*
   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 /* 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.
```

Fig.5.1 shows the test environment of Fig.4.5 in Chapter 4 filled with non-absorbing, isotropically scattering gray medium with a scattering coefficient of 0.1. For this 10 million particles were traced (It may be recalled that 1 million particles were used for non-participating medium). The volume embedding the whole environment was broken into a total of 15625 small volume elements. The total time for simulation was 27:24hrs on a DRS 6000 workstation.

5.3 Implementation Strategy

To implement the above algorithm it is necessary to choose an appropriate computational strategy and a good data structure. The main issues which need consideration are:

Sampling Methods: The algorithm includes a number of steps dealing with the sampling of the source for position and direction of the emitted particle, sampling for type of interaction with volume or surface, and sampling of direction of the re-

Figure 5.1: A complex 3D scene engulfed in smoke.

flected/scattered particle path. Computational methods for many of these sampling steps have been discussed in detail in Chapter 4.

Pathlength Computation: In a participating medium, in addition to finding the surface interaction point with the nearest surface along the particle propagation direction, it is also necessary to find the point of interaction inside the volume. The algorithm for computing the volume interaction pathlength has already been sketched in Section 5.1. The volume structure assumed is very similar to the one used in a ray tracing acceleration method for computing ray-surface interaction, namely the *Spatial Enumeration technique*[26]. 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. If such fineness is used, both for acceleration of ray-surface intersection computations and computation of volume interaction pathlength, then it 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. A two-level volume subdivision technique has been adopted. They are: 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:
(i) associates with each cell a list of surfaces and a list of participating volume elements whose bounding extents intersect the *cell*, and
(ii) 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 tracing both the *surface interaction pathlength* and *volume interaction pathlength* are computed simultaneously which as the reader will notice is a slight variation in the 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 ($\mathit{pathlength\_measure}+K_{voxel}\Delta S_{voxel} > 0$)
     then /* volume interaction point reached.*/

     $\mathit{volume\_interaction\_pathlength} =$

     $\mathit{cumulative\_pathlength} - \mathit{pathlength\_measure}/K_{voxel}$

     else

     $\mathit{pathlength\_measure} = \mathit{pathlength\_measure}+K_{voxel}\Delta S_{voxel}$

     $\mathit{cumulative\_pathlength} = \mathit{cumulative\_pathlength}+\Delta S_{voxel}$

   - If either the nearest surface of intersection is found or the volume interaction point is reached then goto step (4).

4. Sample interaction distribution function to decide on the interaction event.

5.3.1 3D-DDA

We have used the incremental 3D-DDA algorithm for determining the voxel list both for particle tracing and for rendering discussed later in Section 5.5. This algorithm is described in detail below(Fig.5.2):

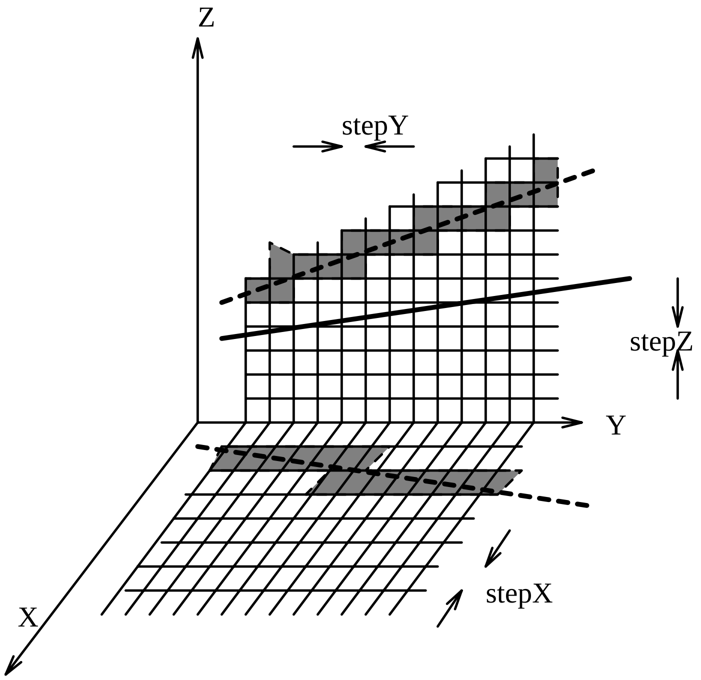

Figure 5.2: 3D-DDA geometry.

Let *ray_entry* be the entry point of the packet in the Box.

Let *tmin* be the entry distance and *tmax* be the exit distance

of the packet starting from the packet origin.

Let *x_subdivision*,*y_subdivision*,*z_subdivision* be the volume

subdivisions in X, Y and Z directions respectively.

Let $x2voxel()$, $y2voxel()$ and $z2voxel()$ be the functions

which return the voxel index along X, Y and Z direction respectively.

Let $tMaxX$, $tMaxY$ and $tMaxZ$ be the distances along the ray to reach

the nearest voxel from the start point along X, Y and Z

directions respectively.

Let *tDeltaX*, *tDeltaY*, *tDeltaZ* be the distances that need be traveled

to cross one voxel along X, Y and Z directions respectively.

Let *stepX*, *stepY* and *stepZ* be the increments required

to reach the corresponding next voxel.

$x = x2voxel(ray\_entry.x);$

```
if ($x == x\_subdivision$) decrement($x$);
if ($ray\_direction.x < 0.$){

        $tMaxX = tmin + (voxel2x(x) - ray\_entry.x)/ray\_direction.x;$
        $tDeltaX = voxel\_size.x/(-ray\_direction.x);$
        $stepX = -1;$

    }
else if ($ray\_direction.x > 0.$){

        $tMaxX = tmin + (voxel2x(x+1) - ray\_entry.x)/ray\_direction.x;$
        $tDeltaX = x/ray\_direction.x;$
        $stepX = 1;$

    }
else{

        $tMaxX = HUGE;$
        $tDeltaX = 0.;$

    }
Similarly compute :

        $tMaxY$,$tDeltaY$,$stepY$ for Y direction $tMaxZ$,$tDeltaZ$,$stepZ$
            for Z direction

    while(1){

        /* Invariant :

                $tMaxX$, $tMaxY$, $tMaxZ$ represent the distance
                    to the nearest voxel from the current voxel.
                    The minimum of the three would give the first
                    voxel pierced by the ray.
            */
        add_to_voxel_list($< x, y, z >$)
        /*
                Choose smallest of $tMaxX$, $tMaxY$, $tMaxZ$. In-
                    crement the index along that axis and update
                    the $tMax$ by the corresponding $tDelta$. If by
                    the process the ray is completely outside the
                    volume then all the voxels have been traversed
                    and the process is stopped.
            */
        if (($tMaxX < tMaxY$) and ($tMaxX < tMaxZ$)) {
                $x+ = stepX$;
                if ($tmax < tMaxX$) break;
                $tMaxX+ = tDeltaX$;
            }else if ($tMaxZ < tMaxY$) {
                $z+ = stepZ$;
                if ($tmax < tMaxZ$) break;
                $tMaxZ+ = tDeltaZ$;
            } else {
                $y+ = stepY$;
                if ($tmax < tMaxY$) break;
                $tMaxY+ = tDeltaY$;
            }
    }
```

5.4 Modelling Participating Volumes

As can be seen in the simulation algorithm discussed in the previous section 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 modelling is currently a very active area of research and any of the volume modelling 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 the following volume modelling primitives have been incorporated in this implementation.

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.

Volumes bounded by Quadric Surfaces: These model a homogeneous medium enclosed within quadric surfaces. Each volume is specified by its canonical quadric form 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.

These volume modelling primitives have been used in order to create the following test environments:

1. The room of Chapter 4 depicted in Fig.4.5 filled with all pervading volume (Fig.5.1).
2. The leaves of the tree modelled with 161 small spherical volumes (Fig.5.3).
3. A flame or a gaseous emitting volume modelled using a dataset (Fig.5.4).

5.5 Rendering

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

Figure 5.3: A tree modelled with participating volumes.

Figure 5.4: An emitting volume.

- 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 radiance 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 (Eq.2.10) stated earlier in Section 2.6.1.

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

$$L \;=\; L_v \;+\; L_s$$

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

$$L_v \;=\; \int_0^{S_{far}} dL_{volume} e^{-\int_0^S K_u du}$$

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.
L_s, the attenuated surface contribution, is the radiance due to the nearest surface along the view direction and is given by

$$L_s \;=\; L_{nearest\_surface} e^{-\int_0^{S_{far}} K_S dS}$$

In the absence of any surface along the view direction $L_{nearest\_surface}$ 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 [64] is:

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

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})

[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.

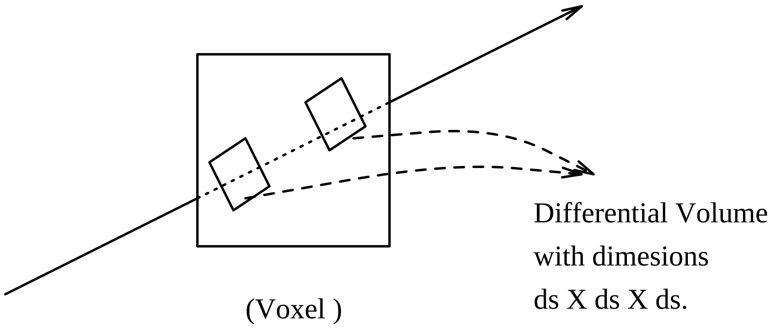

Figure 5.5: Differential volumes within a voxel.

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 (see Fig.5.5) the expression for dL_{volume} can now be written in terms of dS as

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

Substituting the value of dL_{volume} in the equation for L_v we get

$$L_v = \int_0^{S_{far}} e^{-\int_0^S K_u du} \frac{E_{voxel}}{4\pi V_{voxel}} dS$$

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

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

where N is the 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

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

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 voxels along the ray using 3D-DDA algorithm.
    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. $Radiance_\lambda = Radiance_\lambda + i_{nearest\_surface} e^{-sumopacity_\lambda}$
```

5.6 Efficiency Improvement

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 few it contributes a nonzero multiple of its energy. The methods discussed below try to increase the number of nonzero contributions made by the sample particle to the elements of the environment by allowing a fractional contribution of its energy 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.

5.6.1 Forced Interaction

The interaction of the light inside a participating medium is governed by Bouguer's equation(Eq.2.10). This equation gives the factor by which the radiance changes after traveling a distance S inside a participating volume. In Section 5.1 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 fraction of particles exiting the volume without interacting is equal to the above mentioned factor. The actual 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 contribute towards the brightness of the volume elements, this sampling procedure is likely to introduce errors in the final illumination computation.

The method of *Forced Interaction*[43] avoids this sampling problem by forcing the sample particle to interact with each of the volume elements it is passing through, in the process loosing 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 Bouguer's equation if W is the energy associated with the particle entering the volume then the energy of the particle leaving the volume must be $We^{-opacity}$ 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 this particle must interact inside the volume, that the interaction function in a participating medium is exponential in nature, and that the extinction coefficient, K, is constant inside the volume. If S is the length of the particle trace 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 causes 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 used for random number generation, and 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.5.6 shows the visual improvement of the simulation result obtained by using the forced collision method Fig.5.6(a) over the result obtained without forced collision Fig.5.6(b). In both the illustrations, the number of primary particles chosen in forced collision method has been such that the total simulation time using either of the methods remains the same.

Figure 5.6: Results from forced collision and normal simulation.

5.6.2 Absorption Suppression

Light interacting with an element of the environment gets absorbed or 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 the case of interaction with the surface and by the scattering albedo in the case of interaction with the volume. These properties have been used to define a discrete *pdf* of two events which is sampled to decide on the type of interaction for each interacting particle. 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*. Thus this process can introduce errors into the illumination results of an individual surface or volume element.

The *Absorption Suppression*[43] 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, particle tracing 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, as before, one can use unbiased terminating techniques like Russian Roulette to remove a particle whose weight falls below a certain threshold. The usefulness of this method in improving the simulation in a nonparticipating environment has already been demonstrated. Similar improvement has also been noticed for the participating medium.

5.6.3 Particle Divergence Method

There is another sampling step in our simulation which may introduce errors into the simulation results because of the problem of insufficient sampling. This step is the sampling for the outgoing direction for the reflected or scattered particles. 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 traced 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.

5.7 Remarks

We have shown how the particle tracing method for physically simulating the propagation of light is easily extended to deal with complex environments including participating volumes. However, it is not without its disadvantages. In spite of the efficiency improvement techniques discussed above the number of particles that need to be traced for computing the illumination accurately in a complex environment can be prohibitively large. This is particularly true when the environment includes participating volumes of high opacity. A primary reason for this is that many particle traces are not necessarily effective when it comes to computing illumination with reasonable accuracy and could actually be wasteful tracing of the particles. A naive simulation of the physical model of light using light particles results in particle paths which are solely determined by the probability distribution functions that are used in various sampling steps of the simulation process. Many of these particle emissions and the paths traced may not in any way make a significant difference to illumination computations. For example during the course of simulation many particles may interact with an object even after the object's illumination has reasonably stabilised. Similarly many particles may be interacting with objects which are not very relevant to the illumination computation. For example surfaces which are never visible and/or do not illuminate other visible surfaces in any significant manner.

It is clear that if we have to reduce wasted particle emissions and wasteful particle tracings then we have to change the *pdf*s that we use in the simulation so that all particles originate and get distributed in the most useful manner. However, we have to tread this with some care. So far the *pdf*s that we have used in the simulation have been directly derived from the physical models that accurately reflect the optical behaviour of objects in the environment. And as such the illumination that we compute from the simulations conform to the actual physical behaviour of light in that environment. Any change in these *pdf*s would imply that they no longer conform to the physical behaviour of objects. These changes are a must if we have to increase the efficiency of particle tracing. What we therefore really need is a sound mathematical basis which provides for the use of modified *pdf*'s in the simulation, enables the computation of illumination from the simulation results and yet does not deviate from the physical model of light. In the next chapter we introduce and derive the potential equation which forms the mathematical basis for all illumination methods based on the shooting strategy.

Chapter 6

Potential Equation : The Mathematical Basis for Particle Tracing

In Chapter 3 we discussed in detail how various illumination computation methods based on the gathering strategy can be viewed as techniques for obtaining solutions to the radiance equation. We also described progressive radiosity as a deterministic analytic technique based on the shooting strategy. Clearly our particle tracing, described in the previous two chapters, is a non-deterministic technique (a random walk process) also based on the shooting strategy. While the two shooting strategy methods cannot be directly viewed as providing solutions to the radiance equation, it must be recognised that there must exist a similar underlying equation providing the mathematical basis for these methods. Since the prime purpose is the same, i.e., illumination computation, it too must be an integral equation whose solution is required for obtaining flux. Having implemented the particle tracing simulation algorithm and obtained very encouraging results for global illumination computation in complex environments, the mathematical basis for the method was investigated in depth. And the potential equation has been formulated. The *potential equation* and the *radiance equation* together form an adjoint system of equations and provide the mathematical framework for all known approaches to illumination computation. Using the mathematical handles provided by the adjoint system of equations and the random walk model, a number of biasing schemes have been explored for improving the computation of flux estimation.

Of particular significance is the scheme to use an approximate potential value as the biasing function for directing a majority of the random walks through regions of importance in the environment thus reducing the variance in the estimates of flux in these regions. A simple implementation of this scheme has also been carried out. This chapter describes the results of the above investigations in detail.

6.1 The Adjoint System of Illumination Equations

We know that the illumination of any point of a surface in a complex 3D environment is due to the emission of light from that point (if any) and/or due to the reflection from that point of the light received through all the incoming hemispherical directions around that point. This fundamental concept forms the basis for the derivation of the adjoint system of illumination equations. To simplify our discussion, we have restricted our attention to environments containing only opaque solid objects. However, in no way should this assumption be considered as a limitation of the discussed framework. Illumination of environments containing transmitting objects can also be easily explained within the given framework.

6.1.1 Radiance Equation

The general expression for the outgoing radiance is given by Eq.2.8. For conciseness we will drop the parameter λ and implicitly assume dependence on λ. Further we will also drop the subscript o from the radiance as we shall be referring only to outgoing illumination. With this the radiance equation takes the following form:

$$L(x, \Theta_x) = L_e(x, \Theta_x) + \int_{\Omega_x} f_r(x, \Theta_x, \Theta_y) L(y, \Theta_y) \cos\theta_x d\omega_x \tag{6.1}$$

where $L(x, \Theta_x)$ is the outgoing radiance at point x,
$L(y, \Theta_y)$ is the outgoing radiance at point y visible to x along the direction Θ_y,
$L_e(x, \Theta_x)$ is the radiance due to the emission at point x,
θ_x is the cone angle of the incoming direction,
$d\omega_x$ is the differential solid angle around the incoming direction.

6.1.2 Potential Equation

Because of the optical properties of surfaces, which for the present discussion is primarily reflection, the light emitted from any surface in any direction can illuminate many other surfaces of an environment. Alternatively we can say that a surface can be illuminated by lights placed anywhere in the environment. The placement of the lights will of course determine how brightly or how dimly lit that surface is. This phenomenon can be elegantly captured by the notion of a potential associated with every point and direction in the environment. We shall describe a simple experiment to make the concept of potential easier to understand.

For the purpose of illumination computation an environment is generally described in terms of the geometry of its surfaces and their optical properties such as reflection, transmission and emission. To start with, consider an environment completely specified except that its emission characteristics are omitted. Position some hypothetical light detectors in this environment such that the outgoing illumination from any surface point and direction gets registered in one and only one detector. In other words each detector exclusively sees some directional emission of some region of a surface (Fig.6.1). The detectors are hypothetical and in no way affect the flow of light. Next take a hypothetical point source with highly directional emission, emitting a unit amount of luminous flux in any particular direction. If we position this light source at a surface point in some orientation, it is clear that some or all of the hypothetical detectors will register some amount of luminous flux passing through them. Let us concentrate only on one of these, say the k-th detector and note the flux received by that detector because of the placement of the hypothetical emitter. Carry out this exercise for all possible orientations of the hypothetical emitter at that point and at all other surface points of the environments. In the process we would have collected flux values for the k-th detector as a function of all the points and directions of the environment. We will call this as the illumination potential function as this function captures the potential capability of every point and every direction around that point, in illuminating the region on which the k-th detector is focused. Let us denote this function as W_k. Other

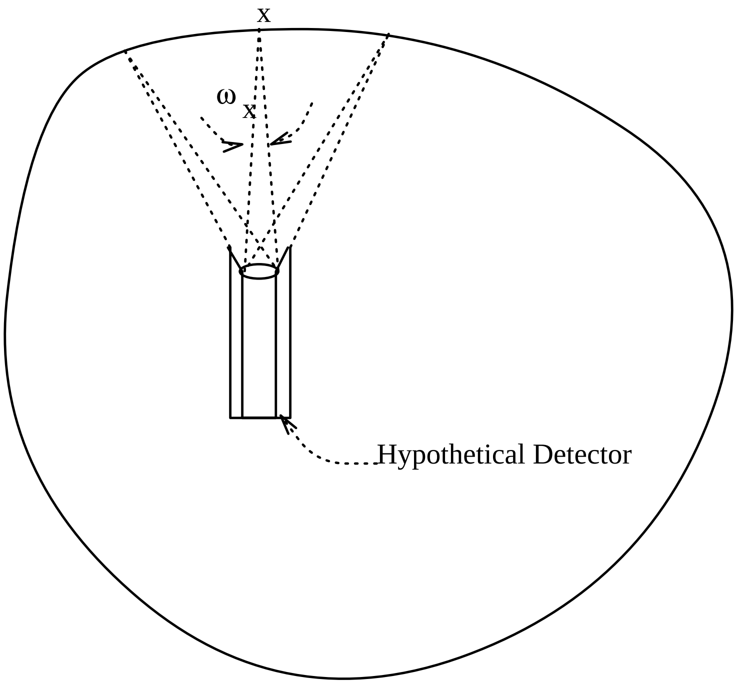

Figure 6.1: A hypothetical detector.

detectors would similarly define potential functions, say W_i.

Next we shall derive an expression for such a function. Let H_k denote the set of all points x over which the k-th detector is focused. Similarly let $\mathcal{D}_k$ denote the set of all directions made by these points with the aperture of the k-th detector. Then we define a function g_k as follows:

$$g_k(x, \Theta_x) = \begin{cases} 1 & \text{iff } (x \in H_k \text{ and } \Theta_x \in \mathcal{D}_k) \\ \\ 0 & \text{otherwise.} \end{cases}$$

Recall that the potential function W_k is the value of light detected by placing hypothetical unit light sources at every surface point in every direction in the environment. Then the immediate contribution of the unit light source placed at (x, Θ_x) in the environment is captured by the function $g_k(x, \Theta_x)$. This is so because the detector would register an immediate unit amount of emission flux only from those emitter positions and orientations, (x, Θ_x), such that $x \in H_k$ and $\Theta_x \in \mathcal{D}_k$ and would register an immediate zero emission flux from all other emitter positions and orientations. We also have to account

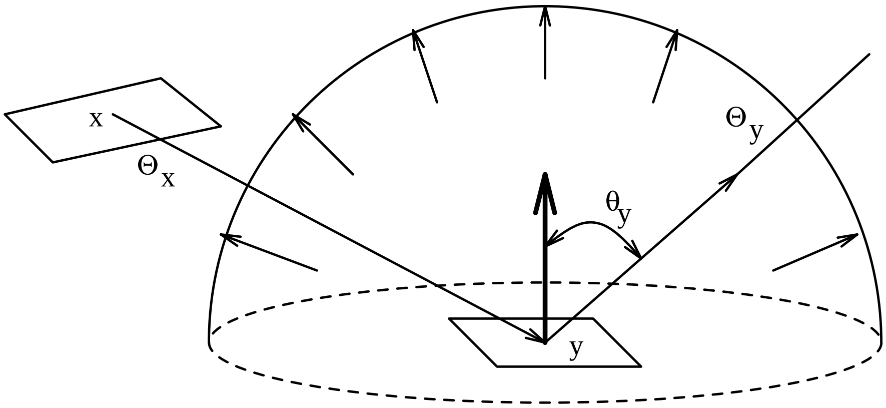

Figure 6.2: Hemispherical directions for outgoing illumination.

for an indirect contribution which is the flux received by the detector due to any number of reflections of the light emitted from this unit light source. For this component we will provide a recursive expression. The emission from the hypothetical emitter at x along direction Θ_x will reach the nearest surface point y and then possibly get reflected. If we take the probability of the whole amount of flux getting reflected in any one of the hemispherical directions Θ_y around y as $f_r(y,\Theta_y,\Theta_x)\cos\theta_y d\omega_y$, where the symbols used are as in Fig.6.2, then its contribution to the indirect component will be this probability times the potential of the point y along Θ_y, i.e. $f_r(y,\Theta_y,\Theta_x)\cos\theta_y d\omega_y W_k(y,\Theta_y)$. The indirect component is then the cumulative result of this expression obtained over the outgoing hemisphere around y, i.e. $\int_{\Omega_y} f_r(y,\Theta_y,\Theta_x)W_k(y,\Theta_y)\cos\theta_y d\omega_y$. The complete expression for the potential function is therefore given by:

$$W_k(x,\Theta_x) = g_k(x,\Theta_x) + \int_{\Omega_y} f_r(y,\Theta_y,\Theta_x)W_k(y,\Theta_y)\cos\theta_y d\omega_y \tag{6.2}$$

We may derive this same equation from a different point of view. If we assume the k-th hypothetical detector to be an emitter of a hypothetical substance called importance and assume that the importance is transported in the environment exactly as light, then due to this emission there will be incidences of importance at various points of the environment. The importance incident at any surface point of the environment can be attributed to the incidences from the detector and from the surrounding points of the environment.

Let us define potential, $W(x,\Theta_x)$, to be the importance per unit projected area per unit solid angle incident at the surface point, x, along the incoming direction, Θ_x. This potential can be expressed as the sum of the potential coming along Θ_x from the detector, and from a point, y, of the environment which is visible to x along Θ_x. The component directly coming from the k-th detector may be expressed by a visibility function, $g(x, \Theta_x)$, which evaluates to 1 if the detector sees the point x along Θ_x and 0 otherwise. The quantity coming from y will be due to the reflection of the potential incident on it from the incoming hemispherical directions, i.e.

$$\int_{\Omega_y} f_r(y, \Theta_x, \Theta_y) W(y, \Theta_y) cos\theta_y d\omega_y$$

Thus the full expression for the potential is

$$W(x, \Theta_x) = g(x, \Theta_x) + \int_{\Omega_y} f_r(y, \Theta_x, \Theta_y) W(y, \Theta_y) cos\theta_y d\omega_y$$

If we now view the outflow of importance from the detector as the inflow of light into the detector then also the same equation for potential will remain valid. Only the incoming directions, Θ's, will now have to be interpreted as outgoing directions and vice versa. Seen from this point of view the potential, $W(x, \Theta_x)$, can be given a physical interpretation of being the potential capability that (x, Θ_x) has towards illuminating the detector.

If we look back at the Eq.6.1 for the radiance equation, we find a striking similarity in the forms of these equations. However it must be noted that in Eq.6.1 the integration is over the incoming hemisphere around x whereas in Eq.6.2 the integration is over the outgoing hemisphere around y, where y is the surface point visible to x in the direction Θ_x.

6.1.3 General Potential Equation

Like the general radiance equation (Eq.2.11), the potential equation can be generalised for environments with participating volumes. From the definition of potential, Eq.6.2, the potential of (x, Θ_x) towards the k-th detector is a sum of a *direct component*, $g(x, \Theta_x)$ and an *indirect component*. This indirect component accounts for all the scattering and reflection events of the light along the direction Θ_x starting from x.

To account for the interaction along the direction Θ_x it is necessary to determine the probability of interaction. From Bouguer's law the probability of the light interacting before traveling a pathlength S is $1 \; - \; e^{-\int_0^S K(u)du}$. Since this is the cumulative probability of the light interacting at every point along the path from 0 to S we get the following expression for the cumulative distribution function.

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

where $p(u)$ is the probability of interaction at distance u. By differentiating the integral equation we arrive at the the expression for $p(u)$ as:

$$p(u) = K(u)e^{-\int_0^u K(z)dz}$$

The general potential equation takes the form:

$$W_k(x, \Theta_o) = g_k(x, \Theta_o) + \int_0^S K(s)e^{-\int_0^s K(z)dz} \left[\int_{\ominus_y} T(y, \Theta_o, \Theta_{in}) W_k(y, \Theta_o) cos\theta_y d\omega_y \right] ds \tag{6.3}$$

where $K(s)e^{-\int_0^s K(z)dz}$ is the probability of light interacting at the distance s from the point x along the directions Θ_x, $T(y, \ldots)$ is the transition probability and $\ominus_y$ represents all possible transition directions at the point y.

We now proceed to find the relationship between the radiance and the potential equation.

6.1.4 Duality

Here we show that Eq.6.1 and Eq.6.2 are duals of each other for the purpose of computation of flux. Duality means that either equation may be used.

Most often in illumination computation one is interested in computing flux from a small region in a small spread of directions. For example: in image rendering the colour of a pixel is assigned by computing the radiance from all the surface points visible to the eye through that pixel and in a spread of directions made by each such point with the aperture of the eye. Similarly in the computation of view independent global illumination of a diffuse environment by radiosity based methods one is computing flux from a small surface patch in the hemispherical direction.

Expression of this flux using the Radiance equation will therefore be an integral of the form:

$$\Phi_k = \int_{position \ \ spread} \int_{direction \ \ Spread} L(x, \Theta_x) \cos \theta_x d\omega_x dx$$

If we assume that this flux represents the flux received by the k-th hypothetical detector then we can use the earlier defined function $g_k(x, \Theta_x)$ which evaluates to 1 in the limits of the integration and 0 everywhere else, and rewrite the above equation as follows:

$$\Phi_k = \int_A \int_{\Omega_x} L(x, \Theta_x) g_k(x, \Theta_x) \cos \theta_x d\omega_x dx \tag{6.4}$$

To get an expression for the above discussed flux using the potential function we will remove the hypothetical source and complete the environment description by adding the actual emissive characteristics to some of the surface points thus defining a function L_e which is zero everywhere except at the positions belonging to emissive surfaces. By introducing L_e we get the emission radiance at (x, Θ_x) to be $L_e(x, \Theta_x)$ and hence the emission flux leaving (x, Θ_x) is $L_e(x, \Theta_x) \cos \theta_x d\omega_x dx$. The potential of (x, Θ_x) towards the k-th detector is $W_k(x, \Theta_x)$. Then the flux received by the k-th detector due to the actual emission at (x, Θ_x) will be $W_k(x, \Theta_x) \times L_e(x, \Theta_x) \cos \theta_x d\omega_x dx$. Thus the expression for the total flux received by the k-th detector will be

$$\Phi_k = \int_A \int_{\Omega_x} W_k(x, \Theta_x) L_e(x, \Theta_x) \cos \theta_x d\omega_x dx \tag{6.5}$$

To sum up

- We have given two different equations, Eq.6.4 and Eq.6.5, to express the same quantity Φ_k using two different functions L and W_k.

- Eq.6.4 and Eq.6.5 are similar in form and so also are the Eq.6.1 and Eq.6.2 for L and W_k.

- Eq.6.1, Eq.6.2, Eq.6.4 and Eq.6.5 together form a closed system.

We will write again all these four equations together to highlight the above mentioned points.

$$\int_A \int_{\Omega_x} L(x, \Theta_x) g_k(x, \Theta_x) \cos \theta_x d\omega_x dx = \Phi_k = \int_A \int_{\Omega_x} W_k(x, \Theta_x) L_e(x, \Theta_x) \cos \theta_x d\omega_x dx$$

$$L(x,\Theta_x) = L_e(x,\Theta_x) + \int_{\Omega_x} f_r(x,\Theta_x,\Theta_y)L(y,\Theta_y)\cos\theta_x d\omega_x$$
$$W_k(x,\Theta_x) = g_k(x,\Theta_x) + \int_{\Omega_y} f_r(y,\Theta_y,\Theta_x)W_k(y,\Theta_y)\cos\theta_y d\omega_y$$

The equations satisfying above mentioned properties are said to form an adjoint system.

One may wish to solve Eq.6.4 or Eq.6.5 to compute Φ_k. In the subsequent sections we will discuss solution methods for computing this flux using either of the equations.

6.2 Analytical Solution for a Diffuse Environment

Because of their inherently complex nature it is not possible to analytically solve Eq.6.4 or Eq.6.5. However, simplified forms of these may be amenable to analytical solutions[1]. We shall derive a simplified version of Eq.6.5 by making the same assumptions as used in radiosity:

1. The environment is a collection of a finite number, say N, of small uniformly diffuse patches.

2. As the radiance from any point of any such uniformly diffuse patch is $1/\pi$ times the flux per unit area we shall compute this total flux leaving that patch in all the hemispherical directions.

3. The solution is carried out in an enclosure, i.e. the hemispherical direction around any point in the environment is assumed to be covered by one or more of the patches of that environment and every patch, j, may be assumed to occupy a solid angle, ω_j (which may be zero) in the hemisphere over any surface point.

To derive the analytical approximation of Φ_k using the potential function we introduce the notion of a *hemispherical potential* of patch i and denote it by $\mathcal{W}_{k,i}$. $\mathcal{W}_{k,i}$ is obtained as the average potential of the points of the patch in any hemispherical direction. If the patches are sufficiently small this hemispherical potential may be assumed to be independent of the position on each patch. The expression for this hemispherical

[1] The simplified version of Eq.6.4 is already in use in the full matrix radiosity method.

potential is derived below:

$$\begin{align*}
\mathcal{W}_{k,i} &= \frac{1}{\pi}\int_{\Omega_{x_i}} W_k(x_i,\Theta_{x_i})\cos\theta_{x_i} d\omega_{x_i} \\
&= \frac{1}{\pi}\int_{\Omega_{x_i}} \left[g_k(x,\Theta_x) + \int_{\Omega_y} f_r(y,\Theta_y,\Theta_x) W_y(y,\Theta_y)\cos\theta_y d\omega_y\right]\cos\theta_{x_i} d\omega_{x_i} \\
&= \frac{1}{\pi}\int_{\Omega_{x_i}} [g_{k,i} + \pi f_r(j)\mathcal{W}_{k,j}]\cos\theta_{x_i} d\omega_{x_i} \\
&= \frac{1}{\pi} g_{k,i}\int_{\Omega_{x_i}}\cos\theta_{x_i} d\omega_{x_i} + \int_{\Omega_{x_i}} f_r(j)\mathcal{W}_{k,j}\cos\theta_{x_i} d\omega_{x_i} \\
&= g_{k,i} + \sum_{j=1}^{N} f_r(j)\int_{\omega_{ij}} \mathcal{W}_{k,j}\cos\theta_{x_i} d\omega_{x_i} \\
&= g_{k,i} + \sum_{j=1}^{N} f_r(j)\mathcal{W}_{k,j}\int_{\omega_{ij}} \cos\theta_{x_i} d\omega_{x_i} \\
&= g_{k,i} + \sum_{j=1}^{N} f_r(j)\mathcal{W}_{k,j}F_{ij} \tag{6.6} \\
&= g_{k,i} + \sum_{j=1}^{N} f_r(j)F_{ij}\left[g_{k,j} + \sum_{l=1}^{N} f_r(l)\mathcal{W}_{k,l}F_{jl}\right] \\
&= g_{k,i} + \sum_{j=1}^{N} f_r(j)F_{ij}g_{k,j} + \sum_{j=1}^{N} f_r(j)F_{ij}\sum_{l=1}^{N} f_r(l)\mathcal{W}_{k,l}F_{jl} \\
&= g_{k,i} + f_r(k)F_{ik} + \sum_{j=1}^{N} f_r(j)F_{ij}f_r(k)F_{jk} + \ldots \tag{6.7}
\end{align*}$$

Using this hemispherical potential we can derive the simplified expression for the flux over the k-th patch as follows:

$$\begin{align*}
\Phi_k &= \int_A\int_{\Omega_x} W_k(x,\Theta_x)L_e(x,\Theta_x)\cos\theta_x d\omega_x dx \\
&= \sum_{i=1}^{ns}\int_{A_i}\int_{\Omega_x} W_k(x,\Theta_x)L_e(x,\Theta_x)\cos\theta_x d\omega_x dx \\
&= \pi\sum_{i=1}^{ns} L_e(i)\mathcal{W}_{k,i}\int_{A_i} dx \\
&= \pi\sum_{i=1}^{ns} L_e(i)A_i\mathcal{W}_{k,i} \tag{6.8} \\
&= \pi\sum_{i=1}^{ns} L_e(i)A_i\left[g_{k,i} + f_r(k)F_{ik} + \sum_{j=1}^{N} f_r(j)F_{ij}f_r(k)F_{jk} + \ldots\right] \tag{6.9}
\end{align*}$$

where ns is the total number of source patches in the environment.

And this is how the computation proceeds in the progressive refinement approach for the radiosity computation[12]. Just as the full matrix radiosity solution is an approximate solution to the rendering equation, the progressive radiosity method is analogously an approximate solution to the potential equation.

6.3 Monte Carlo Methods and Random Walks for General Solution

We shall next discuss a general solution method for computing flux using Eq.6.4 or Eq.6.5. Basically we have to carry out a multidimensional integration, given Eq.6.4 or Eq.6.5 The integration is further complicated by the fact that a component of the integral, in turn, has the form of an integral equation of the second kind.

As already mentioned in Chapter 3 the solution of multidimensional integrations are best carried out by Monte Carlo quadrature techniques. For this we have to find a suitable *pdf*. In both equations i.e. Eq.6.4 and Eq.6.5 we have a predefined known function each, g_k and L_e respectively. We can convert these functions to some constant times a normalised function. That means

$g_k(x, \Theta_x)$ may be converted into $\mathcal{G}_k \times G_k(x, \Theta_x)$

where $\mathcal{G}_k = \int_A \int_{\Omega_x} g_k(x, \Theta_x) \cos\theta_x d\omega_x dx$

and emission function $L_e(x, \Theta_x)$ may be converted to $\mathcal{E} \times S(x, \Theta_x)$

where $\mathcal{E} = \int_A \int_{\Omega_x} L_e(x, \Theta_x) \cos\theta_x d\omega_x dx$.

Then the quadrature process starts by sampling G_k and S. Any standard sampling technique may be used. For each such sample L and W_k are evaluated. As said earlier, L and W_k are integral equations of the second kind. We shall use the random walk technique for solving these integral equations. We discuss below in detail the use of random walk techniques for the evaluation of radiance(L) and potential(W_k).

A random walk or a Markov chain is basically a sequence of states. In order to formulate it we must define the set of all possible states (discrete or continuous) of the system, a starting state and the transition probability function (T) for transition from one state (s) to another (s') such that $\int T(s \to s')ds' \leq 1$. The next state is chosen from the current state by sampling this transition probability function. The transition kernel is said to be normalised if $\int T(s \to s')ds' = 1$, whereas it is said to be subcritical when $\int T(s \to s')ds' < 1$. In a subcritical situation, the probability of $(1 - \int T(s \to s')ds')$ is taken as the probability of no transition (absorption) from state s. Hence a random walk with a subcritical transition kernel is bound to terminate in a finite number of steps as every particle has eventually to be absorbed in some state. Whereas any random walk with a normalised kernel can go on for ever. In the latter case, the walk has to be terminated by some other external criterion. It is natural to expect that all environments would include some absorption. Thus the environment for illumination computation is always subcritical with $f_r(x, \Theta_x, \Theta_y)\cos\theta_x$ as the transition kernel for solving the radiance equation. Similarly $f_r(y, \Theta_y, \Theta_x)\cos\theta_y$ is the subcritical transition kernel for solving the potential equation. A straight forward evaluation of L or W_k using a random walk results in paths consisting of a finite number of steps. The states in our environment are the continuum of surface positions and hemispherical directions around each such surface position. The starting states are sampled from the respective *pdf*s i.e. $G_k(x, \Theta_x)$ or $S(x, \Theta_x)$.

The evaluation of Eq.6.4 may be carried out by drawing n samples from the *pdf*, $G_k(x, \Theta_x)$, and evaluating L by the random walk for each sample (x_i, Θ_{x_i}). If the i-th random walk starting from the state $(x_{i_0}, \Theta_{x_{i_0}})$ covers m_i steps, $(x_{i_1}, \Theta_{x_{i_1}})$,$\ldots$ $(x_{i_{m_i}}, \Theta_{x_{i_{m_i}}})$, then the radiance is estimated from this walk by the formula given below:

$$< L(x_i, \Theta_i) >= L_e(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} L_e(x_{i_k}, \Theta_{x_{i_k}}) \tag{6.10}$$

From this, the estimate of Φ_k can be arrived by averaging over n such random walks.

$$\begin{aligned}
\Phi_k &= \int_A \int_{\Omega_x} L(x, \Theta_x) g_k(x, \Theta_x)\cos\theta_x d\omega_x dx \\
&= \mathcal{G} \int_A \int_{\Omega_x} L(x, \Theta_x) G_k(x, \Theta_x)\cos\theta_x d\omega_x dx \\
&= \mathcal{G} \times \frac{1}{n}\sum_{i=1}^{n}\left[L_e(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} L_e(x_{i_k}, \Theta_{x_{i_k}})\right] \qquad (6.11) \\
&= \mathcal{G} \times \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{m_i} L_e(x_{i_k}, \Theta_{x_{i_k}})
\end{aligned}$$

This method of evaluating Φ_k by first sampling the G_k function is the essence of Kajiya's *path tracing*[38] method.

Similarly the evaluation of Eq.6.5 may be carried out by drawing n samples $(x_{i_0}, \Theta_{x_{i_0}})$ from the source function, $S(x, \Theta_x)$ and carrying out the random walk. A random walk may terminate at the state $(x_{i_k}, \Theta_{x_{i_k}})$ with probability

$$\sigma_{(x_{i_k}, \Theta_{x_{i_k}})} = 1 - \int_{\Omega_y} f_r(y, \Theta_y, \Theta_{x_{i_k}}) \cos\theta_y d\omega_y \tag{6.12}$$

or proceed to the next state $(x_{i_{k+1}}, \Theta_{x_{i_{k+1}}})$ chosen with probability $f_r(x_{i_{k+1}}, \Theta_{x_{i_{k+1}}}, \Theta_{x_{i_k}}) \cos\theta_{x_{i_{k+1}}}$ and so on. For each such sample W_k is estimated from the random walk by the formula given below:

$$< W_k(x_i, \Theta_i) >= g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} g_k(x_{i_k}, \Theta_{x_{i_k}}) \tag{6.13}$$

Once again Φ_k can be estimated from n such walks as follows:

$$\begin{aligned}
\Phi_k &= \int_A \int_{\Omega_x} W_k(x, \Theta_x) L_e(x, \Theta_x) \cos\theta_x d\omega_x dx \\
&= \mathcal{E} \times \int_A \int_{\Omega_x} W_k(x, \Theta_x) S(x, \Theta_x) \cos\theta_x d\omega_x dx \\
&= \mathcal{E} \times \frac{1}{n} \sum_{i=1}^{n} \left[g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} g_k(x_{i_k}, \Theta_{x_{i_k}}) \right] \qquad (6.14) \\
&= \mathcal{E} \times \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{m_i} g_k(x_{i_k}, \Theta_{x_{i_k}})
\end{aligned}$$

This method of evaluating Φ_k by first sampling the source function is the essence of *particle tracing.*

Of the two solution methods, *particle tracing* is more intuitive as it resembles the physical illumination process. Sampling of the source for a start state may be thought of as the emission of a photon from the source and the transition for simulation of random walks may be thought of as the wandering of the photon in the environment as it gets reflected and scattered by the objects in the environment until it is absorbed. *Path tracing* though not directly related to the physical process is by now well known to the *computer graphics* community. The eye point, p_{eye}, and a random point on the pixel, p_{pixel}, define the direction, $\Theta_o = p_{eye} - p_{pixel}$. This direction along with the nearest surface position along $-\Theta_o$ define the starting state for the random walk. At that nearest surface the ray is absorbed and the walk terminates or is reflected along one of the incoming hemispherical directions, Θ_{in}, by sampling the *brdf* and the walk is continued.

What is more important to note in the discussion so far is that both the random walk processes attempt to solve the same problem and are subject to similar statistical errors which in Monte Carlo parlance is known as *variance.* But one aspect which makes *particle tracing* more attractive for global illumination purposes is that the simulation proceeds by sampling the source function. If we partition the space into a finite number of subregions $sr_1, sr_2, sr_3, \ldots$, then we can locate detectors focused over each of these i.e. formulate equal number of g functions $g_1, g_2, g_3, \ldots$, such that g_i is nonzero in the respective subregion sr_i and zero otherwise. Then each random walk originating from the source contributes towards the estimation of Φ_i for each of the subregions. At the end of the simulation we have the estimates for Φ_i for all the subregions. Whereas in *path tracing* the random walk starts by sampling a particular g_i, for example: directions through a particular pixel. So each random walk contributes towards the estimation of only the Φ_i for that region for which g_i is defined to be nonzero. This is not meant to be understood as saying that the computational efforts required to compute the brightness of a pixel by path tracing and to compute the illumination of all the subregions visible through a pixel by particle tracing are of equal magnitude. One may arrive at a low variance in the brightness estimate of the pixel by tracing a small number of paths whereas it is possible that even after a large number of particle tracings the brightness estimates of a few of the subregions continue to show high variance. However, the difference is worth repeating:

> In particle tracing a single random walk contributes towards the estimation of many Φ_is as against many random walks contributing to a single Φ_i in path tracing.

There have been a number of efforts to combine these approaches and thus derive benefits of both. These have typically come to be known as two pass methods or more generally multi pass methods[11, 33, 52, 66, 71]. In the initial passes simulation proceeds starting from the light sources and estimates are obtained for the flux in different subregions. For example, radiosity[11, 66] or particle tracing[52] is used in the first pass to estimate the flux over diffuse surfaces. Chen et al[11] have an additional pass in which rays are traced from the light sources through non-diffuse surfaces to estimate caustics. In the case of multiple initial passes, care is taken to ensure that the flux computations are non-intrusive. The final rendering pass is always from the eye which is based on the random walk solution for Eq.6.4 with the slight difference from path tracing in that the walks are absorption suppressed and the walk is terminated at a diffuse surface whose illumination computation has already been carried out in the earlier passes.

Knowing the basic solution processes we shall now discuss some strategies for increasing computational efficiency. Most of our discussions will be based on the particle tracing method. However, it must be emphasised that both the solution methods will be equally benefited by these strategies.

6.4 Improved Estimation Strategies

We discuss a few methods based on the following observations (i) each random walk contributes either zero or nonzero values to the estimation of a Φ_i, (ii) in most of the situations of interest, more specifically in the problem of illumination computation of a reasonably complex environment, the fraction of random walks contributing nonzero values towards the estimate of any single Φ_i is small[2]. Applying the law of large numbers stated earlier in Chapter 4, a simple minded approach of improving the estimated result will be to increase the number of random walks. Each random walk requires some amount of computational effort for – sampling the initial state, sampling the transition probability function for moving to the next state and computing the nearest surface along a given direction. So any increase in the number of random walks involves a proportionate increase in computation and must be contained. It can be seen that many random walks may in fact never visit the subregion(s) of interest or may visit

[2] In *particle tracing* it rarely happens that every subregion of the space is visited in a single walk. Similarly in *path tracing* it is also equally rare that every random walk starting from the eye will visit a light source during its walk.

subregions in which there have already been an adequate number of visits and hence resulting in not making any further significant contribution to the flux estimates of those subregions. So the strategy to adopt would be either to transform the basic underlying random walk process or the estimator or both such that each random walk almost always contributes significantly towards the region of our interest.

6.4.1 Next Event Estimation

In this technique the stochastic process under study is kept invariant and the form of the estimator is the only one that is modified[20]. The modification involves the use of the equation below as the estimator of $W_k(x_i, \Theta_{x_i})$:

$$< W_k(x_i, \Theta_{x_i}) >= g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=0}^{m_i} W_k^1(x_{i_k}, \Theta_{x_{i_k}})$$

where

$$W_k^1(x, \Theta_x) = \int_{\Omega_y} f_r(y, \Theta_y, \Theta_x) g_k(y, \Theta_y) \cos\theta_y d\omega_y$$

So instead of Eq.6.14, the estimation of Φ_k is done by the equation given below:

$$\Phi_k = \mathcal{E} \times \frac{1}{n} \sum_{i=1}^{n} \left[g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=0}^{m_i} W_k^1(x_{i_k}, \Theta_{x_{i_k}}) \right] \tag{6.15}$$

Similarly in path tracing the estimator of $L(x_i, \Theta_{x_i})$ is given by the expression:

$$< L(x_i, \Theta_{x_i}) >= L_e(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=0}^{m_i} L^1(x_{i_k}, \Theta_{x_{i_k}})$$

where

$$L^1(x, \Theta_x) = \int_{\Omega_x} f_r(x, \Theta_x, \Theta_y) L_e(y, \Theta_y) \cos\theta_x d\omega_x$$

Once again instead of Eq.6.11, Φ_k is estimated by:

$$\Phi_k = \mathcal{G} \times \frac{1}{n} \sum_{i=1}^{n} \left[L_e(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=0}^{m_i} L^1(x_{i_k}, \Theta_{x_{i_k}}) \right] \tag{6.16}$$

The choice of this estimator is based on the following intuition: If $W_k^1(x, \Theta_x)$ is the direct potential contribution averaged over all possible transitions at x, then by replacing the single sampled contribution of $g_k(x, \Theta_x)$ by $W_k^1(x, \Theta_x)$ the random walk process should converge faster. Similarly if $L^1(x, \Theta_x)$ is the direct contribution from all sources averaged together at x, then by replacing the single sampled contribution of $L_e(x,\Theta_x)$ by $L^1(x,\Theta_x)$ the random walk process should converge faster.

The next event estimation in *path tracing* would mean computing the illumination using a local model at a point of a ray-hit. Though not explicitly mentioned we believe that Kajiya uses this estimator in path tracing as he writes in [38, page 146] "*Calculating emitted ... factors is simply a matter of consulting the ... light models*". We also would like to point out that Chen et al[11, page 167] use a variant of the next event estimation technique in computing the final radiance $I(x,\Theta_x)$. They compute $I_{l,s}(x,\Theta_x)$, a part of $I(x,\Theta_x)$, by Monte Carlo sampling only the source contribution at x.

6.4.2 Biasing

All the methods discussed under this topic transform the mathematical description of the stochastic process with an appropriately modified estimator for Φ in order to make the random walk process converge faster. The illumination process as described in Section 6.1.2 is completely described by the source function and the surface *brdf*s. If we replace them by suitably biased functions then when estimating Φ correctly we must remove the bias by properly compensating for the change. In particle tracing the compensation required is derived below:

Let $S'(x,\Theta_x)$ be the biased normalised source function.

If $T(\Theta_x \to \Theta_y) = f_r(y,\Theta_y,\Theta_x)\cos\theta_y$ is used to denote the transition function then let $T'(\Theta_x \to \Theta_y)$ be the biased transition function.

$$\begin{aligned}
\Phi_k &= \mathcal{E} \times \int_A \int_{\Omega_x} S(x,\Theta_x) W_k(x,\Theta_x)\cos\theta_x d\omega_x dx \\
&= \mathcal{E} \times \int_A \int_{\Omega_x} S'(x,\Theta_x)\left(\frac{S(x,\Theta_x)}{S'(x,\Theta_x)}\right) W_k(x,\Theta_x)\cos\theta_x d\omega_x dx \\
W_k(x,\Theta_x) &= g_k(x,\Theta_x) + \int_{\Omega_y} f_r(y,\Theta_y,\Theta_x) W_k(y,\Theta_y)\cos\theta_y d\omega_y \\
&= g_k(x,\Theta_x) + \int_{\Omega_y} T'(\Theta_x \to \Theta_y)\left(\frac{T(\Theta_x \to \Theta_y)}{T'(\Theta_x \to \Theta_y)}\right) W_k(y,\Theta_y) d\omega_y
\end{aligned}$$

The above equation can be written in a more compact form by defining a multiplication factor f where

$$f(x, y) = \frac{T(\Theta_x \to \Theta_y)}{T'(\Theta_x \to \Theta_y)}$$

The transformed potential equation can now be written as

$$W_k(x, \Theta_x) = g_k(x, \Theta_x) + \int_{\Omega_y} T'(\Theta_x \to \Theta_y) f(x, y) W_k(y, \Theta_y) d\omega_y \tag{6.17}$$

With compensation the flux is now estimated by using the following equation:

$$\Phi_k = \mathcal{E} \times \frac{1}{n} \sum_{i=1}^{n} \frac{S(x_{i_0}, \Theta_{x_{i_0}})}{S'(x_{i_0}, \Theta_{x_{i_0}})} \left[g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} \left(\prod_{l=0}^{k-1} f(x_{i_l}, x_{i_{l+1}}) \right) g_k(x_{i_k}, \Theta_{x_{i_k}}) \right] \tag{6.18}$$

Below we first consider two special cases of this general biasing mechanism. The first is absorption suppression in which only the transition function is biased and not the source. The second is source biasing in which transitions are not biased. Later in Section 6.4.3 we discuss a more general method for biasing using the potential associated with surfaces.

Survival Biasing or Absorption Suppression

As the name implies, in this method the absorption probability at the transition points is reduced (may even be made zero) and as a consequence the random walk stretches to longer distances and the probability of each random walk making a nonzero contribution to the estimation of Φ_is is increased. The absorption probability σ at any state is given by Eq.6.12. Any reduction in this probability can be achieved by appropriate increases in the reflection probabilities. A very convenient method is to scale the reflection probabilities simply by the factor $\frac{1}{1-\sigma}$, consequently making the absorption probability at every state to zero. Thus the compensated estimate can be derived from Eq.6.18 as shown below:

$$\Phi_k = \mathcal{E} \times \frac{1}{n} \sum_{i=1}^{n} \left[g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} \left(\prod_{l=0}^{k-1} (1 - \sigma_{x_{i_{l+1}}}) \right) g_k(x_{i_k}, \Theta_{x_{i_k}}) \right] \tag{6.19}$$

A word of caution is needed here; if the transition probability is changed such that there is no absorption at all then every single random walk will go on for ever without terminating. In practice the walk is terminated when the product term in the above equations falls below some minimum threshold. However this termination process introduces a bias into the estimation. An unbiased termination technique like Russian Roulette may be used to overcome this[2, 52]. It may be recalled that in Chapters 4 and 5 we have made use of this technique for improving the estimates obtained from particle tracing simulations.

Source Biasing

In particle tracing, the emission function, $S(x, \Theta_x)$, plays an important role as every random walk originates at the light source. Any biasing of this function while still keeping the normalisation condition satisfied and the transition probability unaltered will change the form of flux estimation equation from Eq.6.18 to the one shown below:

$$\Phi_k = \mathcal{E} \times \frac{1}{n} \sum_{i=1}^{n} \frac{S(x_{i_0}, \Theta_{x_{i_0}})}{S'(x_{i_0}, \Theta_{x_{i_0}})} \left[g_k(x_{i_0}, \Theta_{x_{i_0}}) + \sum_{k=1}^{m_i} g_k(x_{i_k}, \Theta_{x_{i_k}}) \right] \tag{6.20}$$

where $S'(x, \Theta)$ is the biased source function for sampling.
As we shall see later source biasing provides a simple and convenient mechanism for improving the efficiency of particle tracing.

6.4.3 The Use of Approximate Potential for Biasing

Suppose we wish to bias our random walk process to improve the estimate of some specified region in the environment, say, the *region of importance*. This region of importance could be predefined. For example, in rendering a view of a 3D environment, the set of all visible surfaces could form the region of importance. Alternatively the region of importance could be adaptively defined as the solution progresses. This would imply that the importance of regions would change depending on the values computed from a partial simulation. Biasing of particle tracing must be such that the resulting emissions and transitions must lead most of the the random walks directly or indirectly to the region of importance. Further the computations required by the biasing scheme must be simple and straight forward. One possible scheme is to suitably weight the emission function and the transition probability functions.

The potential towards the region of importance provides an excellent basis for this weighting. Let $S(x, \Theta_x) > 0$. Then we can bias the source function to $S'(x, \Theta_x)$ such that the $S'(x, \Theta_x)$ is much greater than $S(x, \Theta_x)$ for those points, (x, Θ_x), whose potential is higher and $S'(x, \Theta_x)$ is much lower than $S(x, \Theta_x)$ for those points whose potential is lower. We can similarly bias the transition probability i.e. in our case the reflection probability as follows: Consider two directions Θ_1 and Θ_2 in the outgoing hemisphere at point x. Let S_1 and S_2 be the two surfaces nearest to point x along directions Θ_1 and Θ_2 respectively. Denote their potential towards the region of importance by W_1 and W_2 respectively. Without loss of generality assume that $W_1 > W_2$. Then the transition function T at x must be biased such that $T(x, \Theta_1)$ is much greater than $T(x, \Theta_2)$.

There is however one catch to the above biasing scheme. It will work provided we know the value of potential that all surfaces in the environment have towards the region of importance. It is clear that if we can derive the exact potential values then we can also derive the solution for the problem at hand and hence we do not require the simulation. Fortunately for biasing purposes we need not know the exact potential values. It is sufficient to obtain approximate values of this potential, hopefully, with much reduced computational costs. Provided these approximate values maintain their relative ordering they can be effectively used to bias the emission and transition probability functions.

6.5 Computation of Approximate Potential and Biasing

In its general form the potential function is dependent both on positions and directions in the corresponding outgoing hemispheres(spheres) of the points of the surfaces(volumes) in the environment. Similarly the region of importance is defined as a collection of points and corresponding directions. In order to illustrate the use of the potential for biasing we shall make the following simplifying assumptions:

1. The environment consists of ideal diffuse reflecting and emitting surfaces.

2. The medium is non-participating.

3. The region of importance is a set of patches with all the corresponding hemispherical directions included.

4. For biasing we shall only use the direction independent hemispherical potential defined earlier in Eq.6.7.

With the above assumptions the environment can now be defined as being made up of patches, say, $E = \{P_1, P_2, \ldots, P_n\}$, such that the region of importance R is a subset of E and for all $P_i \in E$, $\mathcal{W}_i$ denotes the hemispherical potential that patch P_i has towards illuminating patches of R.

The approximate potential values are easily computed from a particle tracing simulation using a much smaller number of particles, say 5-10% of the total required for a complete unbiased simulation. For the purpose of computing hemispherical potential the following additional information is kept track of:

- the number of particles leaving a patch P_i, i.e. emitted/reflected, say N_i,
- the number of these particles reaching a patch belonging to the region of importance, say M_i.

The ratio $\frac{M_i}{N_i}$ gives us an estimate of the hemispherical potential of patch P_i.

6.5.1 Source Position Biasing using Hemispherical Potential

If $\mathcal{W}_i$ is the hemispherical potential of patch P_i then $\mathcal{W}(x)$ is also the hemispherical potential of point x, where $x \in P_i$. Using $\mathcal{W}(x)$ we can bias the normalised source function $S(x)$. Renormalising the biased source function then gives us the following definition:

$$S'(x) = \frac{S(x) \times \mathcal{W}(x)}{S0}$$

Where $S0 = \int_A S(x)\mathcal{W}(x)dx \approx \sum_{i=1}^{ns} S_i \mathcal{W}_i A_i$ and ns is the number of source patches. This biasing results in an altered distribution of source strength, so that emissions take place more often on emitter patches from which the particles have a higher probability of reaching R, the region of importance. To compensate for this biasing for each particle the brightness contributing strength is multiplied by a factor, f_1. The expression for f_1 is given below:

$$f_1 \;=\; \frac{S(x)}{S'(x)} = \frac{S0}{\mathcal{W}(x)}$$

6.5.2 Direction Biasing using Hemispherical Potential

Direction biasing is used both for choosing the direction for emission and for reflection. In the normal simulation the direction is chosen by sampling the diffuse distribution function. In the biased case both for emission and reflection, the idea is to look around the environment and decide on the direction that has a higher probability of leading the random walk to the region of importance. To understand direction biasing using hemispherical potential it may be worth while to look at Eq.6.6 again, which gives a linear expression for the hemispherical potential, and Eq.6.8 which gives us an expression for flux using the hemispherical potential:

$$\begin{aligned}\mathcal{W}_i &= g_i + \sum_{j=1}^{N} f_r(j)\mathcal{W}_j F_{ij} \\ \Phi &= \pi \sum_{i=1}^{ns} A_e(i) A_i \mathcal{W}_i\end{aligned}$$

If we assume that the simulation is being carried out in an enclosure then the outgoing hemisphere around any point is covered by other surface patches of the environment. Associated with each surface patch is its hemispherical potential. Now using the above equations for particle tracing, the transition of a particle can be carried out by sampling the F_{ij} distribution to choose the patch, say k, and sampling the directions occupied by that patch on the hemisphere to arrive at the direction of flight. This F_{ik} times the approximate hemispherical potential now gives us a measure of the new relative importance, F'_{ik}, of each patch around the point p. Now instead of the distribution of F_{ik}, the distribution of F'_{ik} is used for sampling and choosing the appropriate range of directions. Further directions within that range are sampled to choose at the direction of flight for the particle. The resulting mathematical change to Eq.6.6 is as follows:

$$\mathcal{W}_i \;=\; g_i + \sum_{j=1}^{N} f_r(j) \frac{\mathcal{W}_j}{\mathcal{W}_{j,approx}/\mathcal{W}0} F'_{ij} \tag{6.21}$$

where $\mathcal{W}0 = \sum \mathcal{W}_{j,approx} F_{ij}$ and $F'_{ij} = (\mathcal{W}_{j,approx}/\mathcal{W}0)\, F_{ij}$.

There is one major task to be carried out in implementing this idea which is finding the F'_{ij} distribution around a point p on patch P_i. We have already discussed a simple method of computing approximate potential earlier in this section. The task that remains is to determine the F_{ij} distribution in the outgoing hemisphere of the point p. Though it is possible to do this by carrying out hemicube projections coupled with depth sorting it is impractical to use this method for every transition of a particle. Of course the fact that we need only relative importance of the surrounding patches implies that the exact F_{ij} values are not necessary. Once again any suitable approximation which maintains this relative ordering of F_{ij} would do. We have devised a simple method of obtaining this information from the partial simulation used to compute approximate hemispherical potential. This method is based on the following observation:

> If particles are shot diffusely towards the outgoing hemisphere from a point p of the i-th patch, P_i, then the number of particles reaching the j patch, P_j, visible to this point is proportional to F_{ij}. If N is the total number of particles shot from p of a patch P_i and M is the number directly reaching patch P_j then the ratio $\pi\frac{M}{N}$ approaches F_{ij} as N increases. Obviously if a patch is not visible to the point p then M would be zero and so would be F_{ij}.

To be able to capture this information from an unbiased particle tracing process we have used a very simple data structure. The data-structure is a 2-D array of size $N \times N$. We shall name this data structure as *Vis*. Each row of *Vis* corresponds to an immediate emitter and each column corresponds to an immediate receiver. During the process of particle tracing, for every emission/transition, the array cell corresponding to the row of the source patch and the column of the target patch is incremented. $\pi Vis[i][j]/\sum_j Vis[i][j]$ is then used to obtain the approximation to F_{ij}. This value multiplied with the computed approximate hemispherical potential gives us F'_{ij}. The F'_{ij} of all the N patches results in a discrete distribution of patches for transition, or in other words, directions for emission/reflection. By sampling this distribution we get a patch for transition and by sampling the surface of the patch we get the point of transition. The current point and the chosen point of transition define the direction of flight for the particle. This method however has a problem. A patch may only be partially visible. Hence the chosen point of transition on the patch may give rise to a direction which is hidden from the source point. This problem is avoided by rejection sampling. That is the surface is resampled until we arrive at a proper direction. The biasing algorithm is now given below, assuming that at the point p on patch P_i a direction has to be chosen.

```
1. Compute the discrete distribution of $F'_{ij}$, i.e. $\frac{\mathcal{W}_{j,approx}F_{ij}}{\sum_j \mathcal{W}_{j,approx}F_{ij}}$ of patches around $p$.

2. Discrete sample the above distribution and choose a patch say $k$.

3. do{

         Sample the surface of the patch $k$ and choose the transition point $y$ on
         the $k$-th patch

   }while(transition point $y$ on the $k$-th patch is not visible to point $p$).

4. Choose the interaction.
   If the interaction is not absorption then set $p = y$, $i = k$ and repeat from step 1.
```

For the proper computation of flux the compensation factor which appropriately modifies the brightness contribution of the particle is derived from equation (Eq.6.21) and is as follows:

$$f_2 = \frac{\mathcal{W}0}{\mathcal{W}_{j,approx}} = \frac{\sum \mathcal{W}_{j,approx}F_{ij}}{\mathcal{W}_{j,approx}}$$

We have implemented the above biasing mechanisms and have applied it to a number of cases. The resulting improvements in efficiency have been extremely encouraging. Below we discuss these results in a little more detail. Let us first consider the situation in which R is predefined. The simulation then proceeds in two distinct phases. In the first phase the approximate potential values are computed, while in the second phase

Figure 6.3: Scene for importance biasing with predefined R.

the computed potentials are used to bias the source and transition functions and a biased simulation is carried out to obtain global illumination information in the environment. We demonstrate the improvements due to biasing towards a predefined R by using a simple environment, a view of which is shown in Fig. 6.3. The vertical wall on the left extreme has been defined as the region of importance. The wall has been divided into 32×16 patches. Fig.6.4 shows the particle incidence map on the wall with a total of 100,000 particles traced in the simulation. Fig.6.4(a) is the map for normal simulation and Fig.6.4(b) is the map for biased simulation. As one can see visually there is appreciable improvement. The quantitative figures are as follows: 46,462 incidences in the unbiased simulation and 352,922 incidences in the biased simulation. The number of samples rejected during the transition biasing is 69,520 giving an over all improvement factor of 4 with equivalent computation effort.

In the above we have assumed that the region of importance, R, is predefined, and that R forms a small subset of the entire environment. The basic strategy has been based on the use of approximate potential values obtained from a small simulation run for biasing and thus directing most of the random walks to R. This situation is typical

Figure 6.4: The plot of particle incidences on the region of importance.

of view dependent illumination computation. On the other hand for view independent illumination computation clearly the whole environment is the region of importance. Biasing techniques that direct random walks to a region of importance therefore are not meaningful. However this biasing mechanism could still be used effectively to improve computational efficiency of the simulation provided we could devise a strategy like the one stated below:

> The region of importance, which to begin with is the entire environment is gradually pruned as the simulation progresses to smaller and smaller subsets of the environment, and biasing is done for each new subset of important regions.

In our work we have been able to devise one such strategy. This is based on the observation that, as the simulation progresses, some of the regions of the environment would have received enough particle incidences so that the illumination estimates due to even more incidences can be said to be reasonably invariant. That is, as far as these regions are concerned the simulation need not be continued. We make use of this in order to reduce the set of important regions, R, by saying that the regions which have received enough incidences are no more of importance. What we need is then the capability to bias again with this R and carry out a further simulation.

The simulation starts with the whole environment as R (equivalent to no biasing). After a reasonable number of particle traces, R is reduced by removing those regions having close to equilibrium illumination values. The simulation is continued after using the earlier simulation results to bias towards the new R. The process is continued until R is empty.

The two important tasks in realising the above strategy are:

1. A method of deciding on when a subregion of R has reached near equilibrium illumination.

2. A method of computing approximate hemispherical potential for all the regions of the environment for the reduced R.

For the present the first task has been carried out by taking a very simple approach. Regions receiving particle incidences above some predefined number are assumed to have reached equilibrium status. Of course in practice this strategy would have to be much more sophisticated and would have to depend on other criteria which enable one to decide that adequate incidences have been registered over a patch. So after each simulation R is scanned and pruned. As the simulation progresses R is redefined many times. It is therefore not possible to compute the hemispherical potential values only once at the start as was done earlier. Instead we store the complete history of the particle traces from the initial unbiased run. Every time R is redefined these traces are scanned and the new hemispherical potential is computed.

The expression for the biased transition function F'_{ij} is the same as before. Because we have successive biasing we shall use $F^{(n)}_{ij}$ instead of F'_{ij} and we get the recursive expression for it below:

$$F^{(n)}_{ij} = F^{(n-1)}_{ij} \frac{\mathcal{W}_{j,approx}}{\sum_j F^{(n-1)}_{ij} \mathcal{W}_{j,approx}}$$

Figure 6.5: Wire frame drawing showing the top view of the Cornell Labyrinth.

For computation of $F_{ij}^{(n-1)}$ one approach is to use the recursive expansion till one reaches $F_{ij}^{(0)}$ (i.e. F_{ij} of our earlier experiment) and use the Vis data structure computed from the initial unbiased run. The other approach is to update Vis in every simulation and extract $F_{ij}^{(n-1)}$ directly from the updated Vis. This latter approach is what we have used. We believe it is more efficient due to the fact that the information in Vis is enriched in each simulation. The compensated strength of the particle at the n-th biasing step then becomes:

$$f_2^{(n)} = \frac{\sum_j F_{ij}^{(n-1)} \mathcal{W}_{j,approx}}{\mathcal{W}_{j,approx}}$$

We have used this scheme for computing the illumination in an environment like a maze similar to the *Cornell Labyrinth*[3] (see Fig.6.5 and Fig.6.6). It has a total of 523 patches, all of more or less the same area. We have chosen 100 as the minimum

[3] The name has been chosen because a similar scene was chosen by the Cornell group in a recent SIGGRAPH presentation on *Importance Driven Radiosity*[67].

Figure 6.6: A rendered view of the Cornell Labyrinth.

Batch Size	Region of Importance		Rejected Samples	Hits
	Before	After		
300000(Normal)	523	205	0	5625
3000(Biased)	205	131	4142	10266
3000(Biased)	131	72	3758	10747
3000(Biased)	72	26	4048	10816
3000(Biased)	26	13	3642	10494
3000(Biased)	13	8	4017	10706
3000(Biased)	8	6	4224	10247
3000(Biased)	6	4	4324	10513
3000(Biased)	4	0	3888	11083

Table 6.1: Biasing improvements for Cornell Labyrinth.

number of particle incidences on a patch after which we assume that the patch has reached equilibrium illumination. For an unbiased simulation if each and every patch had to receive at least 100 particle incidences then the total number of particles that had to be traced in the entire simulation was 27,000,000. In the case of a biased simulation, 300,000 particles without any bias were first traced. The results were used to prune R to result in 205 patches. The particle tracing history of all these 300,000 was stored for computing $\mathcal{W}$ as and when necessary. The subsequent simulation runs were carried out in batches of 3000 particles each. After each biasing run R was updated and a new $F^{(n)}$ computed. Table 6.1 gives some of the statistics from this experiment. In the table the column corresponding to "Rejected Samples" indicates the number of times the position sampling on the patches for choosing a transition direction resulted in hidden transition points. As can be seen efficiency improvement due to biasing is enormous.

6.6 Remarks

The potential equation for illumination is a powerful mathematical tool for illumination computation by what have usually been called forward simulation techniques or what we have been referring to as shooting strategy methods. The most popular of these, progressive radiosity, is an analytic solution to this equation. Monte Carlo quadrature and random walk methods that can be devised for solving this equation are more general, in the sense that, the simplifying assumptions made for progressive radiosity are not any more necessary. The idea of using the potential for biasing and improving the efficiency of the Monte Carlo solution has been used in other disciplines like Neutron Transport. Its application to illumination computation is not only interesting but also very beneficial. The use of illumination computation in Computer Graphics is for imaging and this naturally defines visible regions as being more important. Using the potential for biasing random walks towards these regions of importance has resulted in very high efficiency improvement factors. Similarly the strategy of successively pruning the region of importance, recomputing the bias and carrying out continuously biased simulations has also proved to result in very high efficiency. This inspite of the fact that the biasing scheme that has been devised and implemented is rather simple and straightforward. Certainly one can expect more sophisticated biasing techniques resulting in even more efficient Monte Carlo solutions to the potential equation and its use for illumination flux computation.

Chapter 7

Conclusions and Future Directions

7.1 In Retrospect

There are four major contributions to the general field of illumination computation that have resulted out of the research reported in this thesis. They are:

- A taxonomy of illumination computation methods.
- Particle tracing techniques for global illumination computation.
- The potential equation for illumination computation.
- Demonstration of the practicality of this new class of global illumination computation algorithms.

Below we shall analyse each of these in a little more detail.

7.1.1 A Taxonomy of Illumination Computation Methods

The primary classification of all illumination computation methods is based on the basic light behaviour simulation strategy used in the method, namely, gathering or shooting. Of particular significance is the fact that each of these strategies has a sound mathematical underpinning in the form of an integral equation whose solution gives us the required illumination values. As we know, by now, the radiance equation is the basis for the gathering strategy, while the potential equation is the basis for the shooting strategy. Basically shooting and gathering are one and the same. Light shot from an emitter into a receiver can be easily viewed as light gathered by the receiver from the emitter. However, there exists a subtle difference in the computation methods based on these. It is that a single shooting operation is capable of illuminating many regions of the environment, whereas a single gathering operation is designed to illuminate only one region. So when it comes to computing illumination globally in an environment shooting based methods are inherently more efficient.

A gathering strategy method computes illumination by solving the radiance equation while every shooting strategy method solves the potential equation. These equations are complex integral equations. The nature of the algorithm used in this equation solving process provides a secondary classification of all methods as being deterministic or non-deterministic. Deterministic methods are more efficient as compared to the non-deterministic methods. However, it is not possible to solve such integral equations in their most general form by using only deterministic methods. A large number of simplifying assumptions are essential to be able to formulate a deterministic solution. Though such methods are efficient, their use is limited to highly restricted environments. Non-deterministic methods are quite general. That means, in principle it is not required to make any simplifying assumption for carrying out the solution. However, they can be quite inefficient. The inefficiency may be due to an expensive sampling step (for ex: use of rejection sampling), and/or due to the requirement that a very large number of samples have to be drawn in order to get reasonable accuracy in the result. Multipass methods use both strategies. Usually a deterministic shooting process in the first pass is followed by a non-deterministic gathering process.

Most often the computed illumination values are used for rendering one or more images of the environment as seen from different view points. Rendering a view involves the setting up of viewing parameters and computing of the radiance value coming in through each pixel in the viewing direction. Illumination computation methods can also be classified as being view dependent or view independent methods. In view independent methods the illumination values computed are valid for all views and hence the final step of computing the radiance value through the pixels can be carried out very rapidly for any specific view. With reasonable performance graphics workstations this enables interactive walk-throughs within the 3D environment.

Ray tracing[75] is basically a view dependent illumination gathering method. It computes the illumination only at the points of interest. Basic radiosity[27], also a gathering method, computes view independent illumination while the recently reported importance driven radiosity[67] can compute illumination by gathering in a view dependent manner. The fact that a single shooting operation contributes to the illumination of a number of objects in the environment inherently makes shooting based methods view independent. Thus progressive radiosity[12] and particle tracing are both view independent. Like importance driven radiosity our importance biased particle tracing can also be used for view dependent illumination computations.

It is important to emphasise that true view independence is difficult to achieve. This is because of the fact that the view point and the viewing direction can be completely arbitrary and one should be able to efficiently obtain the illumination for any point in the environment from the stored illumination values. This is possible provided the computed illumination values are such that one can easily reconstruct a computationally simple illumination function that is scalable to any resolution. None of the view independent methods developed so far truly provide this scalability. All the view independent illumination computation methods subdivide the environment into a finite number of regions. For example, surfaces are broken down, to say, smaller polygonal patches. In each of these patches illumination values are accurately computed only at a few discrete points, say, at the vertices of the polygonal patch. Then an interpolation mechanism is used to compute illumination for all other points in the patch. This is valid only if the illumination values are continuous in the given patch and the interpolation mechanism properly reconstructs the illumination function over the patch. For the regions in which this is not true the reconstructed illumination values are bound to be erroneous. Most often a simple bilinear interpolation function is used though of late more sophisticated interpolating functions like the ones in use in areas like FEM are also being tried. The problem of properly discretising the environment for illumination value reconstruction is therefore a very fundamental problem in illumination computation methods.

7.1.2 Particle Tracing Techniques

Particle tracing is the shooting strategy analogue of path tracing[38]. Both are random walk solutions to similar integral equations. In particle tracing the random walk originates at the light sources whereas in path tracing it originates at the eye point. While particle tracing can be viewed as the process of shooting light rays, path tracing, also for that matter ray tracing in general, can be viewed as the process of gathering light particles. However there are some fundamental differences which must be noted.

> Ray tracing and path tracing techniques are basically derived from geometric optics and are based on the principle of reversibility of light behaviour with respect to directions at an interaction point. Geometric optics provides mathematically simplified formulations derived from the two physical models of light, namely the wave model and the particle model. Particle tracing on the other hand directly models a schematic probabilistic description of the interaction of light with matter as given by the particle model of light. The basic particle tracing technique is a simple Monte Carlo simulation of the natural stochastic process describing the propagation of light in a 3D environment.

The other difference stated earlier is worth restating:

> In a particle's random walk every interaction contributes some light to the point of interaction. Thus for computing illumination globally for an environment this is quite efficient. The same is certainly not true of the random walk of a ray in path tracing. Though, in principle, the ray can contribute some illumination information to the points of interaction on its path, the random walk is only capable of gathering directional illumination from the directly visible points.

The ray-object intersection is the basic computational step by which the interaction points of the particle with objects in the environment are determined. Ray-object intersection has been extensively researched and a wide range of methods have been developed for dealing with a variety of geometric shapes[26]. Therefore the method puts almost no restriction on the type of environmental geometry that can be supported. Special purpose hardware developments are also being carried out. A number of acceleration techniques also exist. Particle tracing techniques can benefit from all the new performance improvement methods that are continuously evolving for illumination computation by ray tracing.

The other basic step in particle tracing is the sampling step for determining the start of the particle's random walk and the continuing path to be chosen at each of the interactions during the course of its life. If it is possible to associate the proper *pdf*s with all the emissions and the interactions then there certainly exist methods to sample them. At the worst one has to resort to rejection sampling. Thus particle tracing is capable of simulating light propagation even in the most general kinds of environments.

Particle tracing techniques produce particle fluxes in the environment which are the estimates of the actual light flux. However the problem of using the simulation results to accurately reconstruct the environmental illumination function continues to be an elusive one. In our work we do a simple apriori discretisation of the environment, capture the illumination values for these discrete regions, and then carry out bilinear interpolation for the reconstruction. Optimal discretisation of environments and also more accurate illumination reconstructions are areas which are being actively researched today.

Often radiance is the quantity of interest, specifically for rendering. It is only for surfaces with diffuse emission/reflection behaviour and volumes exhibiting isotropic scattering that it becomes possible to derive the radiance directly from the computed flux. For more general behaviour such derivations are not possible. So, in addition to reconstructing positional distribution, it is essential to reconstruct the directional distribution of the particle fluxes as well. The two-pass approach first proposed in [71] and also used in our implementation solves this problem in a limited manner. In a two-pass approach, though all the light propagation modeling is accurately done, no attempt is made to capture this information during simulation. Instead when required, the directional flux is recomputed from the surroundings with the added expense of gathering from the environment. Unless this extra computation is properly contained the overheads can be prohibitive. Hence this approach can be used only for environments with largely diffuse behaviour of surfaces and isotropic scattering volumes, and with only a few others exhibiting more complex behaviour. Certainly a more comprehensive and tractable solution has to be evolved.

7.1.3 The Potential Equation

So far the radiance equation, in one form or the other, has been the mathematical underpinning of any new illumination computation method that is proposed. Though often it is not this equation, but a highly simplified version that is actually solved. The radiance equation has always been used to justify the correctness and the validity of the mathematical derivations and the subexpressions involved. With the formulation of the potential equation for illumination computation we believe that newer illumination computation methods based on the shooting strategy would use the potential equation, or more generally the adjoint system of equations, as their mathematical basis. Particle tracing has been shown as a random walk solution for computation of light flux using the potential equation. Also, by using the potential equation to provide the mathematical underpinning, it has been possible to develop efficient solutions using importance biasing in particle tracing. The use of *importance*, the discrete version of the potential, in importance driven radiosity[67] is another example of its usefulness in efficiency improvement of an illumination method. These are just some examples of the application of potential equation and many more are sure to follow.

Form factor computation has been a problem addressed by many radiosity based methods. If we may recall, under the assumption of uniform brightness, the form factor is a value that gives a measure of the relative contribution of radiant light that is directly received by one diffuse surface from another diffuse surface. There have been a number of extensions to deal with indirect light transport from one diffuse surface to another via non-diffuse surfaces. The potential in our potential equation is a somewhat similar but a far more general concept. It expresses the potential that any point of the environment has towards the illumination of any other point of the environment in any given direction. In this fashion the potential encompasses within itself all possible radiation transport paths including those being explicitly attended to in form factor computations. Of course solving the potential equation in general is complex and has been the research problem addressed in Chapter 6 of this thesis. All the same, form factors are basically particular case solutions of the general potential equation and hence all their computation methods are solution methods for particular cases of the potential equation. Bringing out this relationship between the much talked about form-factor and the potential equation is important. For, one can see if already existing methods for form-factor computation can be generalised to compute the potential values for importance biasing. Similarly any method derived for computing potential using the potential equation can be suitably simplified and used for extended form-factor computations needed in radiosity methods.

7.1.4 Practical Implementation

Throughout this research extensive practical implementations have been carried out to serve three important purposes:

- As experimental investigations leading to more theoretical formulations, like, Monte Carlo simulation leading to the potential equation.
- For testing and demonstrating the feasibility and computational tractability of the particle tracing algorithms proposed in this thesis.
- For providing the necessary backing to theoretical ideas, like, the use of importance biasing for efficiency improvement in the potential equation solution by random walks.

As a result, a number of particle tracing algorithms have been implemented and tested using the C language in a UNIX environment. Starting with simple algorithms having only the capability to simulate light propagation in a non-participating ideal diffuse environment the software has been gradually updated with powerful algorithms that support variance reduction techniques, non-diffuse emitters, non-diffuse reflectors, participating volumes and importance biasing. All the programs have been ported to execute on Sun[1] Workstations, VAX[2] and DRS-6000[3] machines.

The efficiency of the ray tracing procedure used in the particle tracing has been obtained by using the uniform *spatial enumeration* technique and the 3D-DDA voxel tracking algorithm. Data structures have been designed for efficient sampling of volume interaction. For image rendering we have used a simplified ray-tracing procedure which does the following:

- Extracts radiance from the precomputed illumination values stored at the visible surfaces.

- Gathers illumination for the visible non-diffuse surface points by Monte Carlo quadrature.

- Integrates the volume illumination along the view-direction by stratified sampling.

A number of image formats such as Utah-RLE[69], TIFF[4], Sun-Raster[5], have been supported for being able to display images easily in different environments.

Further, for comparison with particle tracing, radiosity methods supporting full matrix solution and progressive refinement by shooting have also been implemented A rudimentary X-Window based interactive modelling package had also to be developed to enable the modelling of different test environments. While it cannot be claimed that all these are production quality programs, adequate care has been taken during the design development stages to ensure that much of the software is reusable with minimal effort. All the photographs/shaded pictures included in this thesis have been produced using this software.

[1] Sun is the trademark of Sun Microsystems, Inc.

[2] VAX is the trademark of Digital Equipment Corporation(DEC).

[3] DRS is a trademark of International Computers Ltd.(ICL).

[4] TIFF is a trademark of Aldus Corporation.

[5] Sun-Raster is the raster file format used by SUN OS.

7.2 Possible Extensions

At various points of time during the course of this research there have been a number of ideas which, time permitting, could have been explored in greater detail. We give below a list of the more interesting ones.

- **Particle tracing in distributed environments:** The behaviour of each particle is independent of another. This important property makes particle tracing an obvious candidate for parallel implementation or distributed computing. Different processors of the distributed system may independently trace particles and at specific time intervals the results could be averaged. As long as each particle tracing operation is carried out truly in a random fashion it is not even necessary that the machines trace an equal number of particles.

- **Support for more general brdf and scattering functions:** In our implementation support for general surface reflectance and volume scattering has been provided in a restricted fashion. Support for such general optical behaviour could be attempted by accumulating the directional distribution using directional discretisation or spherical harmonics for illumination accumulation.

- **Adaptive environment discretisation for illumination function reconstruction:** During a simulation the manner in which the particles hit an interacting surface is representative of the illumination function over that surface. Based on this information, one approach may be to find a discretisation method, which adaptively subdivides the surfaces as the particle tracing progresses. Yet another approach may be to avoid the discretisation completely but to keep the particle traces and construct the illumination function directly from the particle traces.

- **More efficient implementation of biased particle tracing:** In our implementation of importance biasing we have taken a somewhat simplistic approach to computing the approximate potential and also to selecting the region of importance. Better and faster methods need to be developed.

- **Application of biasing schemes to participating volumes:** The proposed importance biasing schemes are likely to be very useful in reducing the number of particle traces required in the presence of participating volumes in the 3D environment. One possible way would be to direct more and more particle traces to the volume elements by declaring them as the regions of importance. The impact on the number of particles in the simulation and also the accuracy of final results can only be gauged from results obtained from actual implementations.

7.3 Future Directions

The current state of art in illumination computation is such that all the methods have a valid physical basis and can correctly simulate light propagation in order to produce very highly realistic images of 3D environments. Certainly there will be more research in order

- to model more complex optical behaviour including attenuation and scattering effects,
- to provide accurate and efficient solutions for dynamically changing environments, and
- to develop more accurate light behaviour models that can model arbitrary spectral and spatial distributions.

Current simulation technology, however, is far too slow for real use in scientific and design visualisation. There are bound to be improvements with the availability of increased Mflops and hardware accelerators. But the true challenge lies in making the entire process computationally optimal and fast enough for use in 3D visualisation on reasonably powerful machines in general use. Below we outline two potential areas which we believe merit further investigation.

- **Representation and Manipulation of Multidimensional Illumination Functions:** It is well known that illumination is a function defined over a multi-dimensional space, actually the five dimensional space of position and direction.

The purpose of an illumination computation method is to obtain the value of this function at discrete points in this 5D space or to obtain a piecewise representation of this function defined over the surface/volume elements of the environment. The two computational approaches that are currently in use are (1) compute for all the required points (2) interpolate from a few discrete values computed within each small subdomain. The former will certainly produce correct results but undoubtedly at a very high cost. The latter relies heavily on the hunch that in the discretised domain the illumination function can be reconstructed by some simple interpolation scheme. It is generally true that it is possible to subdivide the domain of a function to the extent that within each subdomain the function has a linear approximation within a given tolerance. Clearly arbitrary discretisations will give incorrect results and unrestricted discretisation will require us to pay a very heavy computational price. Assuming a simple bilinear interpolation scheme there have been some attempts based on techniques similar to those of Finite Element Analysis to adequately discretise the 3D environment[9]. Similarly some of the methods for fitting smooth surfaces to scattered data are also being applied to illumination function reconstruction[59, 72]. The development of appropriate mathematical methods for the piecewise representation of multidimensional illumination functions is an area that needs far more research. Optimally subdividing this five dimensional space of a 3D environment for accurately representing the illumination and thus increasing the efficiency of the light simulation process would also need further investigation.

- **Perception Driven Computation:** In Computer Graphics the prime purpose of illumination computation is synthesis of images which enable the visualisation of complex 3D environments. The major emphasis in all computational methods is to simulate light propagation and obtain solutions to the underlying complex integral equations. Certainly, while this is sufficient to synthesise images of high quality, the rigour and accuracy of the final solution does not seem necessary from a perceptual point of view. We may be computing illumination to much higher differential accuracies than are distinguishable by the human eye. What is needed is to carry out the simulation and compute the illumination in such a manner that the synthesised image includes all the necessary perceptual cues that make this image realistic and also virtually indistinguishable from its final more accurate version. An interdisciplinary approach involving the field of Cognitive Psychology, Computer Vision and Computer Graphics would be very interesting for further research.

Appendix A

Radiometry and Photometry for Computer Graphics

To carry out measurements in any branch of science one must know the basic units of that field. In rendering or for that matter in any illumination computation method we are interested in measuring or computing information related to light. In order to carry out such computations knowledge of illumination metrics is essential[1, 18, 46].

A.1 Radiometry

Light is an electromagnetic form of radiation. Radiometry is the science of measuring radiant energy. Hence it is necessary to understand radiometry for carrying out illumination computations. In radiometry there are specific names for the quantities corresponding to energy, energy density, power and power density. They are *radiant energy, radiant energy density, radiant flux, radiant exitance and irradiance.* These quantities deal with energy and power as they are emitted and detected, independent of the wavelength of the radiation involved.

As radiation is basically a flow of energy, radiometry does not deal with static quantities like mass or charge but deals with the measurement of flow of energy. A flow is fundamentally a rate, a quantity of something per unit time. This rate is also called *flux*. The flow of radiation, *radiant flux*, is radiant energy per unit time. The unit for energy is *joule* or *kilowatt hour*. Thus the unit of radiant flux is joule per second or *watt*. The amount of radiant energy can be computed as a time integral of the radiant flux. In what follows the symbol Q is used to represent radiant energy, and Φ to represent radiant flux, i.e. dQ/dt.

Spectral radiant energy, Q_λ, is the radiant energy per unit wavelength interval. Similarly the spectral radiant flux is the radiant flux per unit wavelength interval, i.e. $\Phi_\lambda = d\Phi/d\lambda$.

Radiant Flux Areal Density: This is denoted by $d\Phi/dA$ at a point on a surface and is given as the quotient of the radiant flux incident on or emitted by surface element surrounding the point and the area of the element. Radiant flux density emitted from a surface has been called *emittance.* The preferred term for radiant flux density leaving a surface is *exitance.* Radiant flux density incident on a surface is *irradiance.*

Radiant Exitance: Radiant Exitance, M or Radiosity, B, is the radiant flux leaving the surface per unit area of the surface. It has the unit of *watt per square meter* or W/m^2.

Irradiance: The irradiance, E, is the radiant flux incident on the receiver per unit area of the receiver. Like *exitance* it has the unit of W/m^2. Just for information, *the average irradiance on the surface of the earth due to the emissions from sun is* 1.35×10^3 W/m^2. *This numerical value is called the* solar constant.

Radiation flows outward in a spherical manner from a point source. Thus to deal with this radial flow, concepts such as projected area, solid angle etc. are used.

Solid Angles: Solid angles are the solid geometry equivalent of angles in plane geometry. The unit of solid angle is *steradian* or *sr.* It is defined as the area of the cone, with apex at the center of the sphere, cut out from the sphere of radius 1. The solid angle of the entire sphere is 4π *sr* and that of the hemisphere is 2π *sr.* A small circular area on the surface of a sphere may be approximated by a flat section. Thus the solid

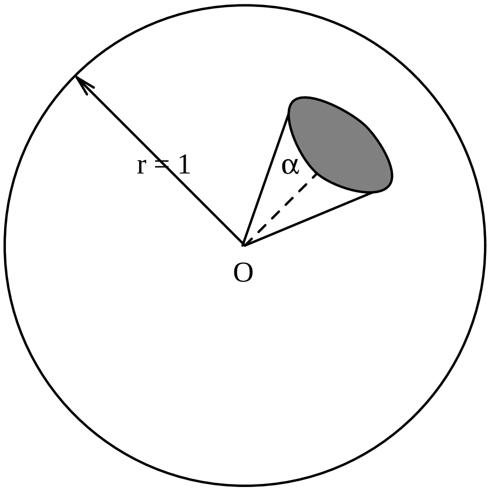

Figure A.1: Solid angle geometry.

angle subtended at the centre of the sphere by this small area may be expressed as:

$$d\omega = \frac{\pi(R\sin\alpha)^2}{R^2} = \pi sin^2\alpha$$

where α is the cone angle.

Projected Area: Projected area is the apparent area of an object seen by an observer from a particular view direction. This projected area, $dA_\perp$, is the actual area, dA, times the cosine of the angle, θ, which the object normal makes with the view direction. i.e.:

$$dA_\perp = dA \ \cos\theta$$

It is clear that the projected area of an object changes with view direction.

Relation between Projected Area and Solid Angle: If we take a small area and try to compute the solid angle that it subtends around a particular point, then

$$d\omega = \frac{dA_\perp}{R^2} = \frac{dA \ \cos\theta}{R^2}$$

Radiant Intensity: Radiant intensity, I, represents the radiant flow from a point source in a particular direction. Thus it is the flux per unit solid angle i.e. $d\Phi/d\omega$ and hence has a unit of watt per steradian, i.e. W/sr.

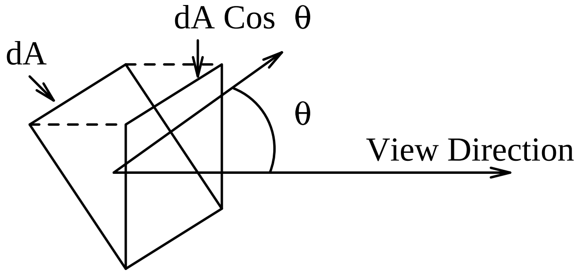

Figure A.2: Projected Area.

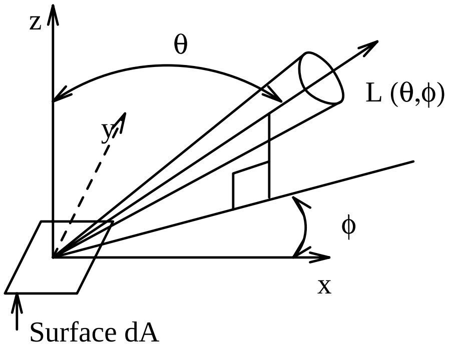

Figure A.3: Radiance geometry.

Radiance: A point emitter is a hypothetical emitting object. Very few actual objects may be approximated as point light sources. These usually are light sources emitting from a very large distance. In real life one deals with area light sources. So to represent radiant flow in different directions from area light sources one uses radiance. *Radiance*, L, along a direction (θ, ϕ) is the radiant flux per projected surface area per unit solid angle centered around that direction. i.e. $L = \frac{d^2\Phi}{d\omega dA \cos\theta} = \frac{dI}{dA \cos\theta}$. The solid angle, $d\omega$, is defined by the cone containing the direction of light propagation with its apex at the point of interest. The projection is on the plane perpendicular to the direction in which the outgoing flow is considered. θ is the angle between the normal to the surface and the given direction. The surface may be real or imaginary and the radiant flux may be leaving, passing through, or arriving at the element of the surface. The unit of radiance is $W/(m^2 sr)$.

Irradiance from Radiance: The radiant flux received by the receiver from a source may be conveniently expressed using radiance. If we consider the emitter with area dA and receiver with area dA', then the total flux received by the receiver may be expressed as

$$d\Phi = L\ dA_{\perp}\ d\omega$$

where $d\omega$ is the solid angle subtended by the receiver at the source. For a small receiver area we may use the solid angle and the projected area relation and rewrite the equation as

$$d\Phi = LdA_{\perp}d\omega = LdA_{\perp}\frac{dA'_{\perp}}{R^2} = LdA'_{\perp}d\omega'$$

where $d\omega'$ is the solid angle subtended by the source at the receiver.

As irradiance is the flux received per unit receiver area, the irradiance at the receiver with area dA' due to an emitter of area dA may be expressed as

$$dq = \frac{d\Phi}{dA'} = L\cos\theta d\omega'$$

This cosine relationship of the irradiance at the receiver with the radiance of the emitter is known as the *Lambert's Cosine Law.*

Using the relationship between irradiance and radiance we can compute for example the solar radiance using the solar constant *as follows:*

The average solid angle subtended by the sun at any point on the earth is 6×10^{-5} *sr.*

The radiance of the sun, L_{sun} *in the direction of the earth is:*

$$L_{sun} = \frac{1.35 \times 10^3\ W}{6 \times 10^{-5}\ m^2 sr} = 2.25 \times 10^7 \frac{W}{m^2 sr}$$

The expression for the irradiance due to a non-differential area emitter will be

$$q = \int_{A'} L\cos\theta d\omega(\alpha')$$

Radiant Energy Density: This is radiant energy per unit volume.

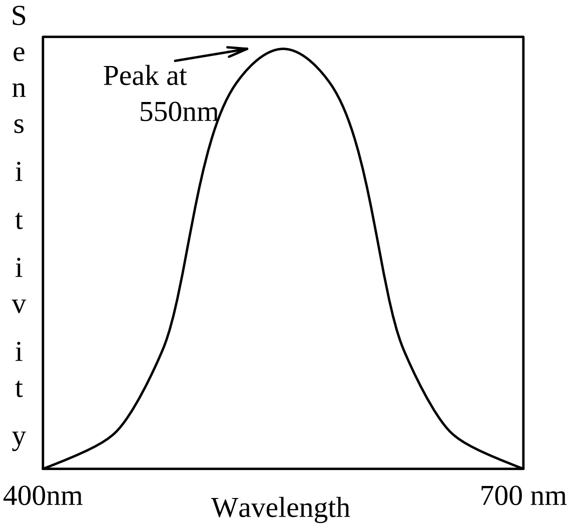

Figure A.4: Sensitivity curve of the human eye.

A.2 Photometry

Photometry is the science of measuring light. Light is the radiant energy capable of stimulating the eye to produce a visual sensation. As already discussed, radiant flux is a physical quantity, where as the light due to a radiant flux is not. The amount of light is dependent on the ability of the radiation to stimulate the eye. The conversion of radiant flux to light involves a factor which depends on the physiological and psychological processes of seeing.

Most of the light measurements are made with respect to the human eye. Radiations with wavelengths too large or too small to be detected by the eye do not play a role in photometry. Thus photometry may be thought of as a specialised branch of radiometry. Different names and units are used for photometric quantities. One can find conversion factors from radiometric units to photometric ones, not vice versa. The sensitivity of the eye varies from wavelength to wavelength. The eye has its peak sensitivity at wavelength $555nm$ which is the same wavelength that the sun has for its peak emission density. The sensitivity curve of the eye falls off at higher and lower wavelengths.

Light: Light is radiant energy, evaluated according to its capacity to produce a visual sensation.

Luminous Flux: Luminous flux, Φ_v, is the rate of flow of light with respect to time. *Lumen* or *lm* is the unit of luminous flux. The lumen is defined as the luminous flux of monochromatic radiation of wavelength *555nm* whose radiant flux is (1/683) *W*. As this wavelength generates the maximal sensation in the eye, larger radiant flux at other visible wavelengths will correspond to 1 *lumen* of luminous flux. The quantity can be expressed as a factor f times (1/683) *W* where f is the reciprocal of sensitivity of the corresponding wavelength relative to the sensitivity at $555nm$.

Luminous Factor or Luminous Efficacy: The sensitivity of the visible wavelengths to the human eye is expressed by *luminous factor or efficacy*. Luminous factor or efficacy of a particular wavelength is the ratio of the luminous flux at that wavelength to the corresponding radiant flux. It is expressed in lumen/watt. The luminous factor or efficacy[1] at $555nm$ is $683\frac{lm}{W}$. The *relative luminous factor* or *luminous efficiency* for a particular wavelength is the ratio of the luminous factor for that wavelength to the value at the wavelength of maximum luminousity i.e. *555nm*. It is dimensionless and is limited to the range 0 to 1.

Quantity of Light: This is the time integral of the luminous flux. *Lumen-hour* or *lm-hr* is the unit of the quantity of light. *Lumen-second* or *lm-sec* is the commonly used unit for expressing the quantity of light delivered by a flash bulb used in photography.

Luminous Intensity: Luminous intensity, I_v, is the solid angular flux density of a point light source in a particular direction, $\frac{d\Phi}{d\omega}$. The *candela* or *cd* is the unit of luminous intensity. One *candela* is one *lumen per steradian.* Since the total solid angle about a point is 4π *steradians* it follows that a point source having a uniform intensity of 1 *candela* has the luminous flux of 4π *lumens.*

[1] Unless otherwise indicated sensitivity measurements and hence luminous factors are defined for photopic vision (i.e. full light colour vision). For scotopic vision (i.e. vision in darkness) the luminous factor for the wavelength of maximum sensitivity is 1754 lm/W.

One candela is the luminous intensity, in a given direction of a source that emits monochromatic radiation of 555nm (frequency 540×10^{12}HZ) and has a radiant intensity in that direction of 1/683 watt per steradian.

Illuminance: Illuminance, E_v or *illumination*, is the areal density of the luminous flux incident on a surface, $\frac{d\Phi}{dA}$. *Lux*, or *lx* is the unit of illumination. One *Lux* is lm/m^2.

Luminous Exitance: Luminous exitance, M, is the areal density of luminous flux leaving a surface at a point. This is the total luminous flux emitted, reflected and transmitted from the surface and is independent of direction.

Luminance: Luminance, L_v, is the photometric equivalent of radiance and is hence a very useful quantity to represent directional luminous flux for an area light source. *Luminance*, L_v, along a direction (θ, ϕ) is the luminous flux per projected surface area per unit solid angle centered around that direction. The surface may be real or imaginary and the luminous flux may be leaving, passing through, or arriving at the surface. *Nit* or *candela per square meter* is the unit of luminance.

A.2.1 Luminance/Radiance the Photometric Brightness

The term brightness usually refers to the strength of sensation that results from viewing surfaces or volumes from which light reaches the eye. For all practical purposes related to human vision, luminance/radiance of an object is taken as the measure of the brightness of the object independent of its dimension and distance. This relationship can be arrived at from the following. If we take Fig.A.5 as a typical setup for imaging using the human eye, the flux received at the pupil of the eye with area S due to the emission from a light source at m with area dA and luminance/radiance L is given by:

$$\Phi = LdA\cos\theta\frac{S\cos\theta}{\mathbf{mp}^2} = \frac{L}{D^2}S\cos^4\theta dA$$

where D is the horizontal distance of the emitter from the pupil of the eye, and θ is the vertical angle with the horizon which the emitter makes at the centre of the pupil. If we assume τ to be the transmission factor of the eye, then this flux times the

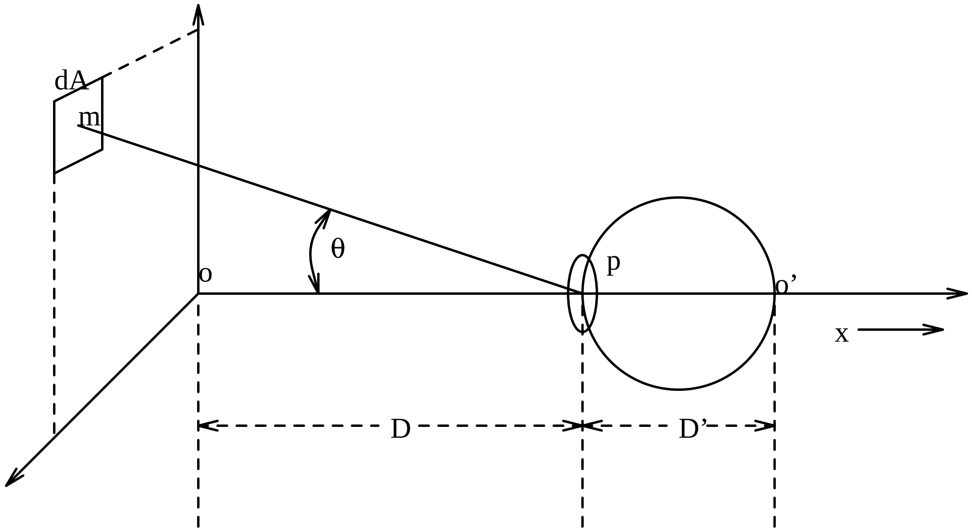

Figure A.5: Imaging at the Eye.

transmittance (τ) of the eye reaches the retina of the eye. If the area of the image at retina is dA' then the illuminance/irradiance of the retinal image is

$$\begin{aligned} q &= \frac{\tau\Phi}{dA'} \\ &= \tau\frac{L}{D^2}S\cos^4\theta\frac{dA}{dA'} \\ &= \tau\frac{L}{D^2}S\cos^4\theta\frac{D^2}{D'^2} \\ &= \tau\frac{L}{D'^2}S\cos^4\theta \end{aligned}$$

If we assume that θ is very small then $cos^4\theta$ approximates to 1. So in practice it is better to ignore $cos^4\theta$. Accommodation of the eye can alter D' slightly, so D' is independent of the object distance. So in all circumstances

$$q = Constant \times L$$

Other things being equal, it follows that retinal illumination and luminance/radiance are always proportional to each other no matter what size and distance of the object. That is why luminance/radiance is the fundamental variable in this thesis.

In image synthesis calculations are done wavelength by wavelength. So the quantities described in the radiometry section are the most applicable. Both in radiometry and photometry the same symbols are used. Therefore to avoid confusion radiometric terms are identified by the subscript e, while photometric (luminous) terms by the subscript v.

Bibliography

[1] *ANSI/IES RP-16-1986 : Nomenclature and Definitions for Illuminating Engineering.* 1986.

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

[3] E. Bahar and S. Chakrabarti. Full wave theory applied to computer-aided graphics for 3-d objects. *IEEE Computer Graphics and Applications*, 7(7):46–60, 1987.

[4] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall Inc., 1982.

[5] D. R. Baum, S. Mann, K. P. Smith, and J. 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.

[6] J. Blinn. Models of light reflection for computer synthesised pictures. *Computer Graphics (SIGGRAPH '77 Proceedings)*, pages 192–198, 1977.

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

[8] W. Bouknight. A procedure for generation of 3d half-toned computer graphics presentations. *Communications of ACM*, 13(9):527–536, 1970.

[9] A. Campbell, III, and D. S. Fussell. Adaptive mesh generation for global diffuse illumination. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4):155–164, Aug. 1990.

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

[11] S. E. Chen, H. E. Rushmeier, G. Miller, and D. Turner. A progressive multi-pass method for global illumination. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):164–174, July 1991.

[12] M. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22(4):75–84, Aug. 1988.

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

[14] R. L. Cook. Practical aspects of distributed ray tracing. In *SIGGRAPH '86 Developments in Ray Tracing seminar notes*. Aug. 1986.

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

[16] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):137–145, July 1984.

[17] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

[18] H. Cotton. *Principles of Illumination*. Chapman & Hall Ltd., 1960.

[19] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Interscience Publishers, 1953.

[20] R. R. Coveyou, V. R. Cain, and K. J. Yost. Adjoint and importance in monte carlo application. *Nuclear Science and Engineering*, 27:219–234, 1967.

[21] F. Crow. Shadow algorithms for computer graphics. *Computer Graphics (SIGGRAPH '77 Proceedings)*, 11(3):242–248, 1977.

[22] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide.* Academic Press, 1988.

[23] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis Horwood, 1987.

[24] R. Feynman, R. Leignton, and M. Sands. *The Feynman Lectures on Physics.* Addison-Wesley, 1963.

[25] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics, Principles and Practice, Second Edition.* Addison-Wesley, 1990.

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

[27] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):212–222, July 1984.

[28] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computing*, C-20:623–628, 1971.

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

[30] R. A. Hall and D. P. Greenberg. A test bed for realistic image synthesis. *IEEE Computer Graphics and Applications*, 3(8):10–20, 1983.

[31] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods.* John Wiley & Sons, 1964.

[32] X. D. He, K. E. Torrance, F. X. Silion, and D. P. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):175–186, 1991.

[33] P. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4):145–154, Aug. 1990.

[34] P. Heckbert. Discontinuity meshing for radiosity. *Third Eurographics Workshop on Rendering*, pages 203–216, May 1992.

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

[36] D. S. Immel, M. Cohen, and D. P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):133–142, Aug. 1986.

[37] J. T. Kajiya. Anisotropic reflection models. *Computer Graphics (SIGGRAPH '85 Proceedings)*, 19(3):15–21, 1985.

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

[39] J. T. Kajiya and B. P. V. Herzen. Ray tracing volume densities. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):165–174, July 1984.

[40] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods. Volume I.* John Wiley & Sons, 1986.

[41] R. V. Klassen. Modeling the effect of the atmosphere on light. *ACM Transactions on Graphics*, 6(3):215–237, 1987.

[42] M. Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9(3):245–261, 1990.

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

[44] N. Magnenat-Thalmann and D. Thalmann. *Computer Animation : Theory and Practice.* Springer-Verlag, 1985.

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

[46] K. D. Moller. *Optics*. University Science Books, 1988.

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

[48] T. Nishita, Y. Miyawaki, and E. 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.

[49] T. Nishita and E. Nakame. Shading models for point and linear sources. *ACM Transactions on Graphics*, 4(2):124–146, 1985.

[50] S. N. Pattanaik. Rad : The radiaosity package. Public Domain Software, available from anon FTP site wuarchive.wustl.edu in graphics/graphics/radiosity/Rad directory.

[51] S. N. Pattanaik. A stylised model for animating bharata natyam: An indian classical dance form. In J. Lansdown and R. A. Earnshaw, editors, *Computer in Art, Design and Animation*. Springer Verlag, 1989.

[52] 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.

[53] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 8(6):311–317, 1975.

[54] P. Poulin and A. Fournier. A model for anisotropic reflection. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4), 1990.

[55] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

[56] R. Y. Rubinstein. *Simulation and the Monte Carlo Method.* John Wiley & Sons, 1981.

[57] H. E. Rushmeier and K. E. Torrance. Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Transactions on Graphics*, 9(1):1–27, Jan. 1990.

[58] H. E. Rushmeier and K. 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.

[59] D. Salesin, D. Lischinski, and T. DeRose. Reconstructing illumination functions with selected discontinuities. In *Proceedings of Third Eurographics Workshop on Rendering*, pages 99–112, May 1992.

[60] Seers, Zemansky, and Young. *University Physics.* Addison-Wesley, 1982.

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

[62] P. Shirley and C. Wang. Direct lighting calculation by monte carlo integration. *Second Eurographics Workshop on Rendering*, May 1991.

[63] P. Shirley and C. Wang. Distribution ray tracing: Theory and practice. In *Third Eurographics Workshop on Rendering*, pages 33–43, May 1992.

[64] R. Siegel and J. R. Howell. *Thermal Radiation Heat Transfer.* Hemisphere Publishing Corporation, 1981.

[65] F. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):187–196, July 1991.

[66] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics (SIGGRAPH '89 Proceedings)*, 23(3):335–344, July 1989.

[67] B. E. Smits, J. R. Arvo, and D. H. Salesin. An importance driven radiosity algorithm. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(2):273–282, July 1992.

[68] J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems.* Addison-Wesley Publishing Company, 1969.

[69] S. G. Thomas. Design of utah rle format. Technical Report TR 86-15, Alpha-1 Project, CS Department, University of Utah, Nov 1986.

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

[71] J. R. Wallace, M. F. Cohen, and D. 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.

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

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

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

[75] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.

[76] L. Williams. Casting curved shadows on curved surfaces. *Computer Graphics (SIGGRAPH '78 Proceedings)*, 12(3):270–274, 1978.

[77] S. Yokoi, K. Kurashige, and J. Toriwaki. Rendering gems with asterism of chatoyancy. *The Visual Computer*, 2(5):307–312, 1986.

List of Figures

List of Tables