Particle Transport and Image Synthesis

James Arvo*
David Kirk*†

*Apollo Systems Division of Hewlett-Packard
300 Apollo Drive
Chelmsford, MA 01824

*†California Institute of Technology
Computer Science 256-80
Pasadena, CA 91125

Abstract

The rendering equation is similar to the linear Boltzmann equation which has been widely studied in physics and nuclear engineering. Consequently, many of the powerful techniques which have been developed in these fields can be applied to problems in image synthesis. In this paper we adapt several statistical techniques commonly used in neutron transport to stochastic ray tracing and, more generally, to Monte Carlo solution of the rendering equation. First, we describe a technique known as *Russian roulette* which can be used to terminate the recursive tracing of rays without introducing statistical bias. We also examine the practice of creating ray trees in classical ray tracing in the light of a well-known technique in particle transport known as *splitting*. We show that neither ray trees nor paths as described in [10] constitute an optimal sampling plan in themselves and that a hybrid may be more efficient.

CR Categories and Subject Descriptors: I.3.7—[**Computer Graphics**]: Three-Dimensional Graphics and Realism; I.3.3—[**Computer Graphics**]: Picture/Image Generation;
General Terms: Algorithms, Graphics
Additional Key Words and Phrases: Boltzmann equation, Monte Carlo, particle transport, radiosity, ray tracing, rendering equation.

1 Introduction

The rendering equation [10] provides a framework in which all current image synthesis techniques can be be viewed as methods of approximation. Both radiosity [7] and ray tracing [18] are examples of approximation because they neglect various optical phenomena in order to yield a reasonable method of solution. An alternative, introduced by Kajiya, is to solve the rendering equation directly via Monte Carlo techniques similar to those developed for neutron transport problems. Such techniques have a long history and have been applied to integral equations of essentially the same form as the rendering equation since the 50's [1].

Kajiya demonstrated the feasibility of this approach in image synthesis by successfully solving the rendering equation for scenes including both specular and diffuse reflectors. Though the level of realism attainable in this way is very high, the cost can be prohibitive due to slow convergence of the Monte Carlo method. Other more efficient approaches have been devised [16,17] but none have completely obviated the need for stochastic approximation without sacrificing certain modes of light transport.

Related statistical approaches have been applied to ray tracing. Cook, et al. [4] described a stochastic sampling technique termed *distributed ray tracing* which provides a means of anti-aliasing as well as simulating effects such as motion blur, penumbrae, depth of field, and fuzzy reflections. Its central idea is that features in the environment which vary in time and space can be sampled stochastically to estimate their contribution to the final image. Both of these paradigms have a great deal in common with Monte Carlo techniques applied to particle transport problems in other fields.

2 Particle Transport

The class of *particle transport problems* consists of those problems which seek to characterize the distribution of idealized particles taking account of their motion and interaction with a medium [5,12,19]. Such problems appear in nuclear engineering as neutron transport [15], in heat transfer as photon transport [13], and in semiconductor device simulation as carrier transport [6]. Many of the equations governing these transport processes ultimately derive from the Boltzmann equation which arose from the kinetic theory of gases. In its simplest form the linear Boltzmann equation can be written as

$$\Phi(P) = S(P) + \int_{\Omega} K(P' \rightarrow P)\, \Phi(P')\, dP' \qquad (1)$$

where P represents particle position, direction, and energy, and $\Phi(P)$ is the density of radiation at P due to emission from the source S as well as contributions scattered into P from all P' [11]. The function K is known as the *scattering kernel*, and the domain of integration, Ω, consists of all positions, directions, and energies. This is a notoriously difficult equation to solve analytically in all but the most trivial problem instances [2,5]. This is true of the rendering equation as well which is essentially a variant of the linear Boltzmann equation. The principal difference is that the scattering kernel is rephrased as a geometry term, g, which accounts for occlusion and inverse square attenuation, and a trivariate scattering term, ρ, whose arguments are surface points (See [10]). The latter encodes the directions of incidence and reflection implicitly through the positions of the source and destination elements relative to the point of reflection. The similarity of the rendering equation to the linear Boltzmann equation suggests that many of the powerful techniques which have been developed for other particle transport problems may be applied to problems in image synthesis.

We note that there are several aspects in which the rendering equation is somewhat more tractable than the transport equations in fields such as nuclear engineering.

1) The particles (i.e. photons) do not influence one another, alter the environment, carry a charge, or replicate via fission. Thus scattering is independent of Φ as well as external forces, making the equation linear.

2) In the absence of participating media, collisions occur only at surfaces. The particles therefore have a relatively large mean free path.

3) We seek only the steady state solution, not transient distributions on the way to equilibrium.

These properties manifest themselves largely in the relatively simple form of the scattering kernel which is comprised of the bidirectional reflectance functions associated with the surfaces. After probabilistically determining a new particle direction at each scattering event the next collision-site along the random walk is completely determined, eliminating stochastic distance calculations. However, there are two respects in which this transport process is made more difficult than typically encountered in other disciplines.

First, the geometry of the simulated environments can be arbitrarily complex. While simulations of reactor cores and semiconductor devices benefit from fairly constrained geometries and exploit special properties of lattices, cylinders, slabs, etc. [12], the trend in computer graphics is to move toward greater and greater scene complexity. This is exemplified by recent work involving billions of geometrical primitives [14]. This can be further complicated by time-dependent scene geometry. Simulation of the resulting motion blur requires time averaging steady-state solutions at intermediate scene configurations.

Secondly, the problem of interest in image synthesis is to compute the intensity of illumination impinging on a single point, the "eye", through small apertures which correspond to "pixels". Analogous situations occur in reactor shielding problems which simulate point radiation detectors [3]. These are inherently more difficult to solve than the typical problems which involve flux averages over volumes.

Many important problems in particle transport do not admit analytic solutions and are also prohibitively expensive to solve via numerical integration due to the high dimension of the *phase space* in which they operate (e.g. three spatial dimensions, two directional dimensions, and an energy dimension). The only recourse for solving these types of problems appears to be Monte Carlo methods which track the behavior of large numbers of particles obeying the prescribed laws of motion expressed as scattering probabilities. Each particle undergoes a sequence of *collisions* or *scattering events* which probabilistically alter its trajectory at each collision-site and contribute to the *history* of the particle. Each particle history, or *random walk*, is used as a statistical estimator of average case behavior. Ray tracing is a mechanism for computing points of collision, and a stochastic ray path [10] is the resulting random walk of a particle. The rendering equation provides a link which allows us to view image synthesis in terms of particle transport. Through this connection we can gain useful insight into the features and limitations of image synthesis techniques.

For example, consider the use of decoupled passes of ray tracing and radiosity to model specular and diffuse modes of transport independently. It has been observed that simply combining the results of these passes fails to account for some important phenomena of geometrical optics [16]. The most obvious example is a caustic formed by specularly transmitted or reflected light falling on a diffuse surface. Both classical ray tracing and radiosity totally neglect this mode of transport, therefore this deficiency cannot be remedied by summing their contributions *a posteriori*. Wallace describes a solution for this particular case of specular-to-diffuse transport, but it is impossible to account for all such sequences of transport as special cases. This phenomenon has been observed in other linear transport problems and is attributed to the fact that equation 1, though linear in the source term, S, is nonlinear with respect to the scattering kernel, K. While the linearity in S allows us to sum the independent contributions made by different sources and wavelengths of light, the analogous decoupling fails when the kernel is partitioned into, for example, $K_{\text{spec}} + K_{\text{diff}}$. A faithful simulation of all modes of transport can only be achieved by coupling them in the solution process.

3 Russian Roulette

The *albedo* of a surface is the probability that an incident particle will be re-radiated after collision [3]. In Monte Carlo simulations this probability is normally used to adjust a numerical *weight* associated with the particle rather than probabilistically terminating the history. This technique, termed *implicit capture* [12], has better statistical properties owing to longer particle histories.

A property of implicit capture is that particle histories can only terminate at surfaces of zero albedo or by *leakage*, that is, by escaping the system. However, it is nearly always impractical to continue tracing a path until one of these conditions is met. Even if we could guarantee the eventual termination of every history, we would spend an inordinate amount of time computing collisions involving particles of negligible weight. One solution is to place a limit on the number of scattering events in a particle history and to ignore all contributions beyond this point. A better solution is to use *weight cutoff* which truncates the particle's history only when its weight falls below some threshold [12]. The idea of using weight cutoff to terminate ray tracing recursion was introduced by Hall [8] and termed *adaptive tree depth control*. Both of these techniques are commonly employed in ray tracing implementations in order to avoid excessively deep ray trees and, in extreme cases, even unending recursion due to opposing mirrors or total internal reflection. The difficulty with this type of policy is that truncation intro-

```
if weight < Thresh then
    begin
    sample s uniformly from [0,1]
    if s < P then terminate path
    else weight ← weight/(1 − P)
    end
```

Figure 1: The Russian Roulette algorithm which is used to terminate particles with insignificant weights without introducing bias. The value P can be any probability less than 1.

duces a systematic bias to the estimator which may become significant if applied to a large number of paths.

Fortunately, this bias can be eliminated by a simple technique known as *Russian roulette* [3,12,15]. According to this technique, once the weight of a particle has fallen below a pre-defined threshold we terminate its history probabilistically, with some given probability, P. If the particle "survives," its weight is increased by a factor of $1/(1-P)$. Let w denote the weight of a particle before playing Russian roulette and let the random variable W denote its subsequent weight. The expected value of W, denoted $E(W)$, is then given by

$$\begin{aligned} E(W) \;=\; & \text{Prob}(\ \textit{termination}\ ) * 0 \;+ \\ & \text{Prob}(\ \textit{survival}\ ) * \frac{w}{1-P} \end{aligned} \tag{2}$$

But the probability of termination is P, and that of survival is $1-P$, so we have

$$P * 0 \;+\; (1-P) * \frac{w}{1-P} \;=\; w \tag{3}$$

which is the the original weight of the particle. On average, then, the particle will have the appropriate weight. We may therefore ignore the majority of the insignificant particles by artificially inflating the contributions of those which survive. Although eliminating the bias in this way does in fact increase the variance slightly, if applied to particles of sufficiently low weight this can be more than compensated for by the additional samples we can collect for the same overall cost. Perhaps more importantly, eliminating the bias guarantees that we will converge to the correct result in the limit if the sample mean converges at all. The Russian Roulette algorithm is outlined in figure 1.

4 Splitting: Paths vs. Trees

Another technique which is commonly used to improve the efficiency of particle transport simulations is *splitting*. While Russian roulette reduces the number of scattering events at the expense of a slight increase in variance, the goal of splitting is to reduce variance by introducing more scattering events. It works by partitioning a single particle into a multiplicity of particles, tracking their diverging histories independently, then down weighting their contributions appropriately. In reactor simulations splitting is used when a neutron encounters a region which is particularly important or of high sensitivity. Though tracking many light weight particles is costly, it is justified if the variance is reduced sufficiently. Because it is used strictly as a variance reduction technique and not as a means of simulating fission, it is applicable to photon transport as well.

In the classical approach to ray tracing introduced by Whitted [18], a single ray can recursively spawn a multiplicity of rays at surfaces which both reflect and transmit light specularly. Cook, et al. [4] generalized this approach by replacing the deterministic branching steps by probabilistic ones distributed over spatial and temporal dimensions. Through the generality of Monte Carlo integration, this allowed a wider variety of optical effects to be simulated with the same number of samples. The resulting method of probabilistic branching is essentially an application of splitting.

As Kajiya observed, however, this approach creates unnecessarily bushy ray trees and expends most of the effort at the leaves (higher generation rays) which make only a small contribution [10]. Though Russian roulette (Sec. 3) can help to limit the depth of these trees by terminating low-weight branches fairly, it does not in itself reduce the bushiness of the tree. Kajiya suggests that it is more appropriate to trace paths instead of trees. At each collision event, exactly one ray is followed by probabilistically choosing one scattering mode to sample from; for example, either the reflected or the transmitted light.

We can compare the two approaches using the *figure of merit* [12] or *efficiency* [11] of the resulting estimators. This measure, which we shall denote by ϵ, is defined by

$$\epsilon = \frac{1}{\sigma^2\,\tau} \tag{4}$$

where σ^2 is the variance of the estimator and τ is the cost associated with drawing a single sample. In this case a sample consists of a complete particle history. At each collision event we wish to sample the incident illumination in such a way that the entire estimator is as efficient as possible. The idea behind path tracing is to use a single particle, thereby reducing τ, which includes the cost of tracing each ray in the environment. This cost can be considerable for complex environments. On the other hand, averaging many particle histories leads to an estimator with a smaller variance, σ^2. Are there any situations in which this reduction in variance outweighs the cost of tracking multiple particle histories? Though it is difficult in general to estimate both τ and σ^2 for any given strategy, we can nevertheless construct examples in which splitting confers a clear advantage.

Consider a particle which encounters N ideal mirror reflectors before reaching a diffuse reflector. If we estimate the incident illumination at the diffuse surface by tracing a single path, obtaining a variance of σ_1^2 at a cost τ, then the efficiency of the entire estimator is

$$\epsilon_1 = \frac{1}{\sigma_1^2(N\alpha + \tau)} \tag{5}$$

where α is the average cost of tracing a single ray in the environment. On the other hand, if we achieve a slightly lower variance, σ_m^2, by splitting into m paths of the same cost after tracing a single path to the diffuse reflector then the efficiency of the entire estimator is

$$\epsilon_m = \frac{1}{\sigma_m^2(N\alpha + m\tau)} \tag{6}$$

To see that splitting can be advantageous in some instances we need only observe that

$$\lim_{N \to \infty} \frac{\epsilon_m}{\epsilon_1} = \frac{\sigma_1^2}{\sigma_m^2} > 1 \tag{7}$$

This shows that for any given α and τ, after a sufficiently large number of mirror reflections splitting into multiple paths is a more efficient strategy than continuing a single path.

Another instance in which splitting is advantageous is when multiple samples reduce the variance significantly. More precisely, if at any point along a path we can employ m samples of equal cost to estimate the incident illumination and achieve a variance σ_m^2 such that

$$\sigma_m^2 < \frac{\sigma_1^2}{m} \tag{8}$$

where σ_1^2 is the variance of a singe sample, then $\epsilon_m > \epsilon_1$ and we have an increase in efficiency. Under certain conditions, such a reduction in variance can be obtained through *sample stratification*, a common Monte Carlo technique in which the domain of integration is partitioned into disjoint regions which are sampled independently.

If the incident illumination at a surface point can be separated into low-variance strata whose mean values differ greatly, then splitting into one path for each stratum will result in a more efficient estimator [9]. Two strata which will often meet these criteria are the intense direct illumination from light sources and the attenuated indirect illumination from the remainder of the environment [10].

While ray trees generally place too much of the computational burden at the leaves, these examples indicate that there exist cases in which trees lead to greater efficiency than a strict application of path tracing. This suggests that a hybrid method can achieve higher efficiency than either strategy alone if inexpensive heuristics for strategy selection are employed.

5 Conclusion

As the level of realism in computer generated images has grown, the underlying illumination models have encountered many of the complications common to other particle transport problems. This is not surprising when one views the rendering equation as a form of the linear Boltzmann equation, a transport equation which has application in many areas of science and engineering. We can exploit this similarity by drawing upon techniques developed for other transport problems and applying them to image synthesis. Fields such as nuclear engineering are rich sources of statistical techniques which are applicable to stochastic ray tracing and to Monte Carlo solution of the rendering equation. As examples, we have discussed the uses of Russian roulette and splitting in this context. Finally, because image synthesis presents additional challenges due to features such as complex scene geometry, techniques developed for image synthesis may also be useful in other domains.

References

[1] Albert, G. E., "A general theory of stochastic estimates of the Neumann series for solution of certain Fredholm integral equations and related series," in *Symposium on Monte Carlo Methods*, edited by M. A. Meyer, J. Wiley, New York 1956, pp. 37-46.

[2] Cercignani, Carlo, "The Boltzmann Equation and its Applications," Springer-Verlag, New York, 1988.

[3] Clark, Francis H., "Methods and data for reactor shield calculations," in *Advances in Nuclear Science and Technology*, No. 5, 1971, pp. 95-183.

[4] Cook, Robert L., Thomas Porter, and Loren Carpenter, "Distributed Ray Tracing," *Computer Graphics*, 18(3), July 1984, pp. 137-145.

[5] Duderstadt, J. J., and W. R. Martin, "Transport Theory," J. Wiley, New York, 1979.

[6] Ferry, D. K., "Semi-Classical Boltzman Transport Theory in Semiconductors," in *Physics of Nonlinear Transport in Semiconductors*, New York, 1979.

[7] Goral, Cindy M., Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile, "Modeling the interaction of light between diffuse surfaces," *Computer Graphics*, 18(3), July 1984, pp. 213-222.

[8] Hall, R. A., and D. P. Greenberg, "A testbed for realistic image synthesis," *IEEE Computer Graphics and Applications*, 3(10), November, 1983, pp. 10-20.

[9] Hammersley, J. M., and D. C. Handscomb, "Monte Carlo Methods," Chapman and Hall, 1964.

[10] Kajiya, J. T., "The Rendering Equation," *Computer Graphics*, 20(4), August 1986, pp. 143-150.

[11] Kalos, M. H., and Paula A. Whitlock, "Monte Carlo Methods, Volume I: Basics," J. Wiley, New York, 1986.

[12] Lewis, E. E., and W. F. Miller, Jr., "Computational Methods of Neutron Transport," J. Wiley, New York, 1984.

[13] Siegel, Robert, and John R. Howell, "Thermal Radiation Heat Transfer," Hemisphere Publishing Corp., Washington DC, 1981.

[14] Snyder, John M. and Alan H. Barr, "Ray Tracing Complex Models Containing Surface Tessellations," *Computer Graphics*, Vol. 21, No. 4, July 1987, pp. 119-126.

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

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

[17] Ward, Gregory J., Francis M. Rubinstein, and Robert D. Clear, "A Ray Tracing Solution for Diffuse Interreflection," *Computer Graphics*, 22(4), August 1988, pp. 85-92.

[18] Whitted, Turner, "An Improved Illumination Model for Shaded Display," *Communications of the ACM*, 32(6), June 1980, pp. 343-349.

[19] Williams, M. M. R., "Mathematical Methods in Particle Transport Theory," J. Wiley, New York, 1971.