# THE RENDERING EQUATION

James T. Kajiya
California Institute of Technology
Pasadena, Ca. 91125

ABSTRACT. We present an integral equation which generalizes a variety of known rendering algorithms. In the course of discussing a monte carlo solution we also present a new form of variance reduction, called Hierarchical sampling and give a number of elaborations shows that it may be an efficient new technique for a wide variety of monte carlo procedures. The resulting rendering algorithm extends the range of optical phenomena which can be effectively simulated.

KEYWORDS: computer graphics, raster graphics, ray tracing, radiosity, monte carlo, distributed ray tracing, variance reduction.

CR CATEGORIES: I.3.3, I.3.5, I.3.7

## 1. The rendering equation

The technique we present subsumes a wide variety of rendering algorithms and provides a unified context for viewing them as more or less accurate approximations to the solution of a single equation. That this should be so is not surprising once it is realized that all rendering methods attempt to model the same physical phenomenon, that of light scattering off various types of surfaces.

We mention that the idea behind the rendering equation is hardly new. A description of the phenomenon simulated by this equation has been well studied in the radiative heat transfer literature for years [Siegel and Howell 1981]. However, the form in which we present this equation is well suited for computer graphics, and we believe that this form has not appeared before.

The rendering equation is

$$I(x,x') = g(x,x')\left[\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx''\right]. \quad (1)$$

where:

| | |
|---|---|
| $I(x,x')$ | is the related to the intensity of light passing from point $x'$ to point $x$ |
| $g(x,x')$ | is a "geometry" term |
| $\epsilon(x,x')$ | is related to the intensity of emitted light from $x'$ to $x$ |
| $\rho(x,x'x'')$ | is related to the intensity of light scattered from $x''$ to$x$ by a patch of surface at $x'$ |



The equation is very much in the spirit of the radiosity equation, simply balancing the energy flows from one point of a surface to another. 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 toward $x$ from all other surface points. Equation (1) differs from the radiosity equation of course because, unlike the latter, no assumptions are made about reflectance characteristics of the surfaces involved.

Each of the quantities in the equation are new quantities which we call *unoccluded multipoint transport* quantities. In section 2 we define each of these quantities and relate them to the more conventional quantities encountered in radiometry.

The integral is taken over $S = \bigcup S_i$, the union of all surfaces. Thus the points $x, x'$, and $x''$ range over all the surfaces of all the objects in the scene. We also include a global background surface $S_0$, which is a hemisphere large enough to act as an enclosure for the entire scene. Note that the inclusion of a enclosure surface ensures that the total positive hemisphere for reflection and total negative hemisphere for transmission are accounted for.

As an approximation to Maxwell's equation for electromagneticseq. (1) does not attempt to model all interesting optical phenomena. It is essentially a geometrical optics approximation. We only model time averaged transport intensity, thus no account is taken of phase in this equation—ruling out any treatment of diffraction. We have also assumed that the media between surfaces is of homogeneous refractive index and does not itself participate in the scattering light. The latter two cases can be handled by a pair of generalizations of eq. (1). In the first case, simply by letting $g(x,x')$ take into account the eikonal handles media with nonhomogenous refractive index. For participating propagation media, a integro-differential equation is necessary. Extensions are again well known, see [Chandrasekar 1950], and for use in a computer graphics application [Kajiya and von Herzen 1984]. Elegant ways of viewing the eikonal equation have been available for at least a century with Hamilton-Jacobi theory [Goldstein 1950]. Treatments of participatory media and of phase and diffraction can be handled with path integral techniques. For a treatment of such generalizations concerned with various physical phenomena see [Feynman and Hibbs 1965]. Finally, no wavelength or polarization dependence is mentioned in eq. (1). Inclusion of wavelength and polarization is straightforward and to be understood.

## 2. Discussion of transport quantities

We discuss each of the quantities and terms of equation (1). This equation describes the intensity of photon transport for a simplified model. $I(x,x')$ measures the energy of radiation passing from point $x'$ to point $x$. We shall name $I(x,x')$ the *unoccluded two point transport intensity* from $x'$ to $x$, or more compactly the *transport intensity*. The transport intensity $I(x,x')$ is the energy of radiation per unit time per

unit area of source $dx'$ per unit area $dx$ of target.

$$dE = I(x, x')\, dt\, dx\, dx'. \tag{2}$$

The units of $I$ are joule/m$^4$sec,

The term $g(x, x')$ is a geometry term. This term encodes the occlusion of surface points by other surface points. If in the scene, $x'$ and $x$ are not in fact mutually visible then the geometry term is 0. On the other hand if they are visible from each other then the term is $1/r^2$ where $r$ is the distance from $x'$ to $x$. Note that an occluding perfectly transparent surface can make $g(x, x')$ to be equal 0. For, in fact, the transparent surface, intercepts the radiation and reradiates it on the other side.

The emittance term, $\epsilon(x, x')$ measures the energy emitted by a surface at point $x'$ reaching a point $x$. We shall call it the *unoccluded two point transport emittance* from $x'$ to $x$. It gives the energy per unit time per unit area of source and per unit area of target. That is,

$$dE = \frac{1}{r^2}\epsilon(x, x')dt\, dx\, dx'. \tag{3}$$

The units of $\epsilon(x, x')$ are joule/m$^2$sec,

Finally the scattering term $\rho(x, x', x'')$ is the intensity of energy scattered by a surface element at $x'$ orginating from a surface element at $x''$ and terminating at a surface element at $x$. We shall call it the *unoccluded three point transport reflectance* from $x''$ to $x$ through $x'$.† The term $\rho$ is a dimensionless quantity. So the energy reaching $x$ is given by

$$dE = \frac{1}{r^2}\rho(x, x', x'')I(x', x'')\, dt\, dx\, dx'\, dx'' \tag{4}$$

We now relate the transport quantities to more conventional radiometric quantites. We shall do this by equating the energy transported by each quantity for the given geometric configuration.

Ordinary radiometric intensity is defined as energy per unit time per unit of projected area of source per unit of solid angle

$$dE = i(\theta', \phi')d\omega\, dx'_p\, dt. \tag{5}$$

To relate these quantities we look at the imaging geometry in figure 1.

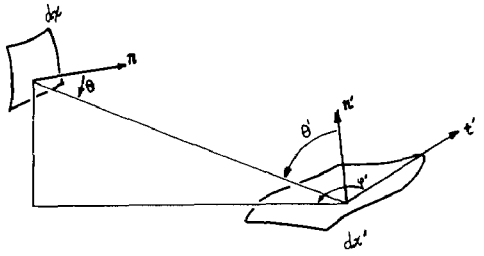



Figure 1. Two point imaging geometry. A frame is attached to each surface element giving a normal, tangent, and binormal vector.

From the figure we obtain

$$\begin{aligned} r &= \|x - x'\| \\ dx'_p &= dx' \cos\theta \\ \cos\theta &= \frac{1}{r}\langle \mathbf{n}, x - x' \rangle \\ \cos\theta' &= \frac{1}{r}\langle \mathbf{n}', x - x' \rangle \\ \cos\phi' &= \frac{1}{r}\langle \mathbf{t}', x - x' \rangle \end{aligned} \tag{6}$$

where:

† This term also covers the transmittance of light through surfaces as well. To simplify the ensuing discussion we will ignore transmission scattering altogether.

- $\mathbf{n}$ is the normal to surface element $dx$
- $\mathbf{n}'$ is the normal to surface element $dx'$
- $\mathbf{t}'$ is the tangent vector to the element $dx'$
- $r$ is the distance from $x'$ to $x$

The solid angle subtended by a surface element $dx$ is the fractional area of a sphere of radius $r$ taken up by the projected area $dx_p$ of $dx$.

$$d\omega = \frac{dx_p}{r^2} = \frac{1}{r^2}\cos\theta\, dx. \tag{7}$$

Thus substituting eq. (7) in eq. (5) we get

$$dE = i(\theta', \phi')\frac{1}{r^2}\cos\theta\cos\theta' dt\, dx\, dx'. \tag{8}$$

Equating eq. (2) and eq. (5) gives the relationship between transport intensity and ordinary intensity

$$I(x, x') = i(\theta', \phi')\frac{1}{r^2}\cos\theta\cos\theta'. \tag{9}$$

The relation between transport emittance and ordinaryemittance is derived likewise. Assuming that there are no occluding surfaces, the energy transmitted by emission from surface element $dx'$ to $dx$ is given by eq. (3). Using the definition of ordinary emittance we can follow exactly the same procedure as above to obtain

$$\epsilon(x, x') = \epsilon(\theta', \phi')\cos\theta\cos\theta' \tag{10}$$

Finally, we relate the transport reflectance to the ordinary radiometric total bidirectional reflectance function$\rho(\theta', \phi', \psi', \sigma')$ from the definition

$$i(\theta', \phi') = \rho(\theta', \phi', \psi', \sigma')i(\psi', \sigma')d\omega'' cos\psi' \tag{11}.$$

Where the imaging geometry appears in figure 2.

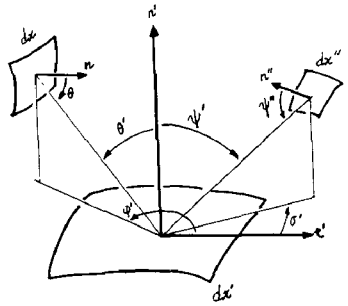



Figure 2. Three point imaging geometry.

From the diagram we obtain in addition to equations (6) and (7), the following

$$\begin{aligned} r'' &= \|x' - x''\| \\ dx''_p &= dx'' \cos\psi'' \\ \cos\psi' &= \frac{1}{r''}\langle \mathbf{n}', x' - x'' \rangle \\ \cos\psi'' &= \frac{1}{r''}\langle \mathbf{n}'', x' - x'' \rangle \\ \cos\sigma' &= \frac{1}{r''}\langle \mathbf{t}', x' - x'' \rangle \\ d\omega'' &= \frac{dx''_p}{r''^2} = \frac{1}{r''^2}\cos\psi'' dx'' \end{aligned} \tag{12}$$

where:

- $\mathbf{n}''$ is the normal to surface element $dx''$
- $r''$ is the distance from $x''$ to $x'$
- $d\omega''$ is the solid angle subtended by surface element $dx''$

Combining eqs.(2),(8),(9),(11), and (12) we obtain the relationship between the unoccluded three point transport reflectance and the or-

dinary total bidirectional reflectance

$$\rho(x, x', x'') = \rho(\theta', \phi', \psi', \sigma') \cos\theta \cos\theta' \tag{13}$$

## 3. Methods for approximate solution

In this section we shall review approximations to the solution of the rendering equation. It appears that a wide variety of rendering algorithms can be viewed in a unified context provided by this equation. During the course of this discussion, many other untried approximations may occur to the reader. We welcome additional work on this area. This territory remains largely unexplored, since the bulk of the present effort has concentrated solely on the solution methods to be presented below.

### *Neumann series*

One method of solving integral equations like eq.(1) comes from a well known formal manipulation, see [Courant and Hilbert 1953]. We rewrite it as:

$$I = g\epsilon + gMI$$

where $M$ is the linear operator given by the integral in eq.(1). Now if we rewrite this equation as

$$(1 - gM)I = g\epsilon$$

where 1 is the identity operator, then we can formally invert the equation by

$$\begin{aligned} I &= (1 - gM)^{-1} g\epsilon \\ &= g\epsilon + gMg\epsilon + gMgMg\epsilon + g(Mg)^3\epsilon \cdots \end{aligned} \tag{2}$$

A condition for the convergence of the infinite series is that the spectral radius of the operator $M$ be less than one. (Which is met in the case of interest to us). A physical interpretation of the Neumann expansion is appealing. It gives the final intensity of radiation transfer between points $x$ and $x'$ as the sum of a direct term, a once scattered term, a twice scattered term, etc.

### *The Utah approximation*

For lack of a better name, we shall call the classical method for rendering shaded surfaces the Utah approximation. In this approximation we approximate $I$ with the two term sum:

$$I = g\epsilon + gM\epsilon_0$$

Thus the Utah approximation ignores all scattering except for the first. The geometry term is by far the most difficult to compute. The Utah approximation computes the $g$ term only for the final scattering into the eye. This is, of course, the classical hidden surface problem studied by many early researchers at the University of Utah. Note that in the second term, the operator $M$ does not operate on $g\epsilon$ but rather directly on $\epsilon_0$. Thus this approximation ignores visibilty from emitting surfaces: it ignores shadows. The $\epsilon_0$ term is meant to signify that only point radiators are allowed. No extended lighting surfaces were allowed. This simplification reduces the operator $M$ to a small sum over light sources rather than an integration over $x''$.

Since that time many extensions have appeared, most notably shadow algorithms and extended light sources.

### *The Ray Tracing approximation*

Whitted [1980], proposed a different approximation:

$$I = g\epsilon + gM_0g\epsilon_0 + gM_0gM_0g\epsilon_0 + \cdots$$

In this famous approximation, $M_0$ is a scattering model which is the sum of two delta functions a cosine term. The two delta functions of course represent the reflection and refraction of his lighting model. The cosine term represents the diffuse component. Note that he gives $g\epsilon_0$: shadows but with point radiators. Whitted's ambient term translates directly to the $\epsilon$ term. Again the operator $M$ can be approximated by a small sum.

### *The distributed ray tracing approximation*

In 1984, Cook [Cook et al 1984], introduced distributed ray tracing. This approximation uses an extension of the three component Whitted model resulting in a more accurate scattering model. This extension necessitated the evaluation of an integral in computing the operator $M$. In this model $M$ is approximated by a distribution around the reflection and refraction delta functions. The innovation that made this possible was the use of monte carlo like techniques for the evaluation. As is well known, the ability to evaluate integrals has widely extended the range of optical phenomena captured by this technique. A proper treatment of the ambient term, however, remained elusive to distributed ray tracing.

### *The radiosity approximation*

In 1984, Goral, Torrance, and Greenburg [Goral et. al. 1984, Cohen and Greenburg 1985, Nishita and Nakamae 1985] introduced radiosity to the computer graphics world. This is a major new rendering technique which handles the energy balance equations for perfectly diffuse surfaces. That is, surfaces which have no angular dependence on the bidirectional reflectance function

$$\rho(\theta', \phi', \psi', \sigma') = \rho_0. \tag{14}$$

The *radiosity* $B(x')$ of a surface element $dx'$ is the energy flux over the total visible hemisphere. It is the energy per unit time per unit (unprojected) area, measured in watts per meter squared. It is defined by

$$\begin{aligned} dB(x') &= dx' \int_{hemi} i(\theta', \phi') \cos\theta' d\omega \\ &= dx' \int_{hemi} \frac{I(x, x') r^2}{\cos\theta} d\omega \\ &= dx' \int_S I(x, x') dx \end{aligned} \tag{15}$$

Thus to calculate hemispherical quantities we may simply integrate over all the surfaces in the scene. So from eq.(1) and (15) we obtain

$$\begin{aligned} dB(x') = dx' \int \Big\{ & g(x, x')\epsilon(x, x') \\ & + g(x, x') \int \rho(x, x', x'') I(x', x'') dx'' \Big\} dx \end{aligned} \tag{16}$$

If there is an occlusion between $x$ and $x'$ then the contribution of the emmitance term is zero. Otherwise the contribution is

$$\begin{aligned} dB_e(x') &= dx' \int \frac{\epsilon(x, x')}{r^2} dx \\ &= dx' \int \epsilon(\theta', \phi') \cos\theta' \frac{\cos\theta dx}{r^2} \\ &= dx' \int \epsilon(\theta', \phi') \cos\theta' d\omega \\ &= dx' \pi \epsilon_0 \end{aligned} \tag{17}$$

Where $\epsilon_0$ is the hemispherical emittance of the surface element $dx'$.

Similarly for the reflectance term, the contribution to radiosity is again zero for an occluded surface. Otherwise we get

$$\begin{aligned} dB_r(x') &= dx' \int \frac{1}{r^2} \int \rho(x, x', x'') I(x', x'') dx'' dx \\ &= dx' \int \frac{1}{r^2} \rho(\theta', \phi', \psi', \sigma') \cos\theta \cos\theta' dx \\ &\quad \times \int I(x', x'') dx'' \\ &= dx' \rho_0 \int \cos\theta d\omega \int I(x', x'') dx'' \\ &= dx' \rho_0 \pi H(x') \end{aligned} \tag{18}$$

Where $H$ is the hemispherical incident energy per unit time and unit area. In this derivation we switched the order of integration and used identities (13),(12), and (14). Now using equations (17) and (18) in (16) we see that the rendering equation becomes

$$dB(x') = \pi[\epsilon_0 + \rho_0 H(x')] dx' \tag{19}$$

Which is equation (4) in Goral et. al. [1984].

Calculating the total integrated intensty $H$ is essential to calculate the final $F_{ij}$ matrix in radiosity. This requires a visibility calculation which may be quite expensive. Since the matrix equation is solved by a number of relaxation steps, it is essentially equivalent to summing the first few terms of the Neumann series: propagating the emitters across four or so scatterers. To use relaxation requires that the full matrix be calculated. Relaxation also gives all the intensities at all the surfaces in the scene. While in certain cases this may be an advantage, it is suggested that the monte carlo method outlined below may be quite superior.

## 4. Markov chains for solving integral equations

The use of Markov chains is perhaps the most popular numerical method for for solving integral equations. It is used in fields as diverse as queuing theory and neutron transport. In fact, the use of monte carlo Markov chain methods in radiative heat transfer has been in use for quite some time, [Siegel and Howell 1981]. In the heat transfer approach, a packet of radiation of specified wavelength is emitted, reflected, and absorbed from a configuration of surfaces in some enclosure. Counting the number of packets absorbed by each surface after a run gives an estimate of the geometric factors whose exact calculation would pose an intractible problem. This is similar to ray tracing a scene from the light sources to the eye. Rather than follow these methods, we will choose to solve eq.(1) more directly going back to an early monte carlo method first put forth by von Neumann and Ulam [Rubenstein 1981].

### *Finite dimensional version*

By way of introduction we first present the method in a finite dimensional context. This simplifies the notation and makes obvious the essential ideas involved. Again we note that this example method may possibly hold many advantages over the currently used relaxation schemes popular in radiosity: intensities at only visible points need be computed, and calculation of the full radiosity matrix may be exchanged for a very much smaller set of selected matrix elements.

Suppose we wish to solve the vector equation:

$$x = a + Mx$$

where $x$ and $a$ are $n$-dimensional vectors, $x$ an unknown, and $M = (m_{ij})$ is an $n \times n$ matrix.

Now from a Neumann expansion we see that for $M$ a matrix with eigenvalues lying within the unit circle, the solution $x$ is given by

$$x = a + \sum_{k=1}^{\infty} M^k a$$

The method evaluates this sum by averaging over paths through the matrix multiplies. That is, it follows a path through rows and columns that comprises an iterated matrix product. For each point in the path we get a row or column which can be indexed by an integer from 1 to $n$.

Construct a probability space $\Omega$ where each point $\omega$ is a path visiting one of $n$ points at each discrete time, viz, $\omega = (n_0, n_1, \ldots, n_k)$ where each $n_i$ is an integer from 1 to $n$. The length $k = l(\omega)$ of the path $\omega$ is finite but otherwise arbitrary and corresponds to an entry in the $k$th matrix power. Each path is assigned a probability $p(\omega)$.

If we wish to calculate the value of one coordinate of $x$, say $x_1$, then we calculate the quantity

$$\hat{x}_1 = (\prod_{i=0}^{l(\omega)} m_{n_{i-1} n_i}) a_{n_{l(\omega)}} \frac{1}{p(\omega)}$$

averaged over all paths $\omega \in \Omega$. Simply taking expected values verifies that this quantity gives the desired quantity.

The probability space of paths is most easily constructed using Markov chains. A (stationary) discrete Markov chain consists of a set of states $X$, and an assignment of a *transition probability* $p(x, x')$ from one state $x' \in X$ to another $x \in X$, and an initial probability density of states $p(x)$. Some subset of states may be designated as *absorbing* in that no transitions out of an absorbing state are permitted.

The probability of a path generated by a Markov chain is simply the the product of the initial state and all the transition probabilities until an absorbing state is reached. So for a path

$$\omega = (x_0, x_1, \ldots, x_{l(\omega)})$$

we have the probability is

$$p(\omega) = p(x_{l(\omega)}, x_{l(\omega)-1}) \cdots p(x_2, x_1) \cdot p(x_1, x_0) \cdot p(x_0)$$

In the finite dimensional case we let the state set of the Markov chain be the set of indices into the vector or matrix, $X = \{1, \ldots, n\}$. Note that although we are allowed wide lattitude in choosing the transition probabilities, they must be positive for the corresponding nonzero entries in the matrix. In the limit our estimate of the solution is quite independent of the probability distribution of the paths. But the rate of convergence to the limit is highly dependent on the manner of choosing the transition probabilities. Section 5 gives a set of new techniques for choosing the transition probabilities.

### *Infinite dimensional solution*

Extending the monte carlo Markov chain method to infinite dimensional equations is straightforward. For the equation at hand, we note that it is a variant of a Fredholm equation of the second kind. The passivity of surfaces in reflecting and transmitting radiation assures the convergence of the Neumann series. We simply replace the state set by the set of points $x$ on a surface. The procedure for calculating the points is thus:

1. Choose a point $x'$ in the scene visible through the imaging aperture to a selected pixel $x$ on the virtual screen.
2. Add in the radiated intensity.
3. For the length of a Markov path do
   3.1 Select the point $x''$ and calculate the geometrical factor $g(x, x')$.
   3.2 Calculate the reflectance function $\rho(x, x', x'')$ and multiply by $\epsilon(x', x'')$.
   3.3 Add this contribution to the pixel intensity.

Note that calculating the emittance and scattering factors is simply a matter of consulting texture maps and lighting models. Calculating

the geometrical factor is, in fact, the ray-object intersection calculation of ray tracing. Note also, that by choosing the next point $x''$ on the Markov path by shooting a ray at an chosen angle and finding the closest intersection point, we in effect perform a powerful importance sampling optimization. That is, we do not bother to calculate the integral for points $x', x''$ which are occluded by another surface because we know the integral will be zero. This is in contrast to the relaxation procedure in radiosity which always takes energy contributions from all surfaces.

## 5. Hierarchical sampling

We now present a number of new variance reduction techniques invented for solving the rendering equation. We hasten to point out, however, that the variance reduction techniques exposed here are of much wider scope. Generally they will have utility in all manner of monte carlo integration problems in which the integrand is particularly difficult. In this situation, the increased overhead beyond previously known methods becomes negligible. We present five methods which take increasing advantage of precious samples of the integrand. All the techniques outlines below were inspired by stratified sampling.

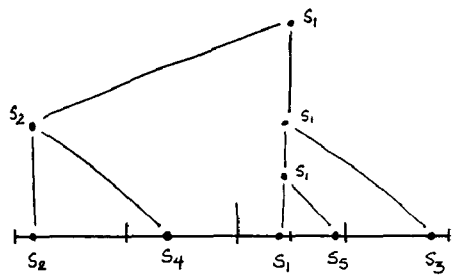
### *Sequential uniform sampling*

The first sampling technique stems from a common sequential sampling strategy. Often samples of the integrand are repeatedly collected until the sample variance of the integral estimate falls below a fixed threshold. This strategy has been shown to be of advantage in [Lee, Redner and Uselton 1985], where many samples were collected at interesting parts of the image while few were collected at uninteresting parts.

Unfortunately, this sequential strategy is incompatiblewith stratified sampling. In the stratified sampling technique, the domain of interest is divided into subcells. Lee, et. al. used a fixed subdivision of 8 cells per pixel and randomly collected samples within each cell. Ideally, better convergence is obtained when one sample per cell is collected, where the cells uniformly divide the domain—this is the so called jitter sampling method, where ordinarily we think of the centers of the cells as forming a lattice. The incompatibility between sequential and jitter sampling arises because a uniform subdivision of the domain is impossible until it is known precisely how many samples will be collected.

Sequential uniform sampling achieves this by keeping a tree of cells of varying sizes. Each time a sample is to be cast, a cell is first chosen and then divided into cells. The old sample of the original cell must lie in one of the new subcells. The new sample is chosen to lie in the opposite cell. A simple example will illustrate this technique.

Suppose we are sampling a unit interval and have already cast 5 samples. The cells chosen with sample points appear thus:




To cast a new sample point we traverse the tree until a leaf cell is encountered. We then split the leaf cell in half and cast the sample into the empty half cell.

REFINE A NODE
1. If the node is an internal node
   1.1 Choose a subnode
   1.2 Refine the chosen subnode
   1.3 Return
2. Else, split the leaf node
3. Propagate the old sample into the subleaf containing it.
4. Cast a new sample in the remaining empty subleaf

How can we assure that the most uniform possible subdivision is computed? One way would be to traverse the sampling tree in breadth first order. Splitting each leaf node at every level before splitting deeper nodes. This strategy produces highly nonrandom sample distributions, essentially scanning across the interval. A better method is to split nodes breadth first in random order. The following criteria effect this strategy

CHOOSE A SUBNODE
1. If either is a leaf choose it.
2. Choose left node if
   level(left)<level(right) and left is balanced
3. Choose right node if
   level(right)<level(left) and right is balanced
4. Choose randomly otherwise.

Note that this strategy will in effect perform a random search throughout the interval, without concentrating on any particular area.

### *The multidimensional case*

The above algorithm is easily extended to higher dimensions simply by using a data structure known as a *k-d tree* due to Bentley [Bentley 1979]. In this data structure, the domain is succesively divided into two halves by a hyperplane perpendicular to successive coordinate axes. Thus for say a unit square, the k-d tree subdivides first along a vertical line, then on the next level down along the horizontal. The uniform subnode choice rules above ensure a uniform subdivision without any modification. Generalization to path spaces is straightforward.

### *Hierarchical integration*

The third version of the above technique takes advantage of the fact that the cells for each sample are recorded with each sample. In this way we may compute a Riemann sum using the volume of the cell and the value of the cast sample as integrand. Yakowitz [Yakowitz et al 1978] has proposed a variant of this method (using the samples themselves as boundary points with no stratification). He has reported a variance of $O(n^{-4})$ in the one dimensional case, and a variance of $O(n^{-2})$ in the two dimensional case. This is in vastly superior to the $O(n^{-1})$ of simple monte carlo. The analysis of our technique is still under investigation, and will appear in a companion paper. But due to the stratification of our samples, early evidence suggests that this is a superior technique for integration.

Each time a leaf cell is split, its contribution to the total integral is divided in half. The new integrand sample is multiplied by the volume of empty cell. After splitting and sampling has occurred, the path from the leaf to the root is traversed, updating the integral stored at each node to be the sum of the integrals of its subnodes. By keeping the integral of nonroot internal nodes we are able to automatically scale the by the density of the samples to maintain a constant measure.

Figure 3 shows the convergence of a two dimensional integral as compared to the conventional monte carlo technique. The value of the integral estimate is plotted versus number of samples cast. The conventional estimator is shown above and the hierarchical integrator is shown below. We are integrating a simple step function on a connected region of the plane.

*Adaptive hierarchical integration*

The fourth elaboration of this technique concerns other criteria besides uniformity of samples in the domain. In this variation, we seek to concentrate samples in interesting parts of the domain and to sparsely sample those areas in which the integrand is nearly constant.

We seek criteria for selecting interesting parts of the tree to undergo further refinement. How can these criteria be included in the algorithm? It is easy to think of the subnode selection rules of the uniform sequential sampler as a way of setting probability thresholds. Choose a uniform random number in the unit interval. The uniform rule calculates a threshold $\phi_u$ which is either 1 or 0 if the rule says to choose left or right subnode. If the rule says to choose randomly, the threshold is set to 0.5.

Now let us calculate a number of threshold's $\phi_1, \ldots, \phi_k$. To take all these threshold functions into account a effective scheme is to form the convex combination of them as the global threshold, that is the global threshold $\phi$ is given by

$$\phi = c_1\phi_1 + c_2\phi_2 + \cdots + c_k\phi_k$$

$$\sum_{i=1}^{k} c_i = 1$$

where $c_i \geq 0$ for every $i$. Each $c_i$ provides a weight for its corresponding threshold function so that the total strategy can undergo tuning.

What are the useful threshold functions? We have found a few, but it is clear that the number of useful criteria left to be discovered is many. Among the threshold functions we have found useful are 1) the uniform sampling threshold; 2) The totally random threshold ($\phi = .5$); 3) The difference of integrals of the two subnodes; 4) A history of the activity of change in this subnode (which may be the variance, or some weighted time history of the integral); and 5) *A priori* functions that can predict where large illumination components will be.

So far our experiments in finding adaptive criteria have not been terribly successful. We have not used adaptation in computing the final images.

Again we note as in the last section, that recording the volumes of the cells in each node automatically provides the normalization that is needed when the sampling distribution is skewed. This is often problematical in adaptive sampling schemes.

Figure 4 shows the unit square subdivided according to criterion 1) and 3) in equal proportion. This is a snapshot of the subdivision when 165 samples have been cast.

*Nonuniform sampling: Importance sampling analogs*

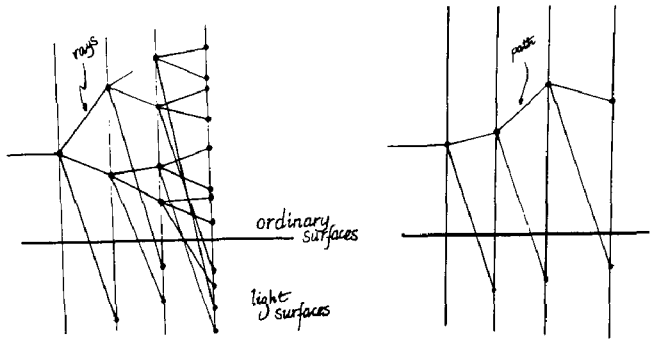
Finally, the fifth technique takes into account importance sampling. Instead of dividing a leaf cell exactly in half, it is possible to divide it along a hyperplane that represents the median of some probability density function. The hyperplane chosen is given by the level of the k-d tree in the second technique. Representing the probability density as an integrated distribution function makes it easy to choose the median hyperplane by a quick binary search: to find the median of a probability density $f(x)$ we simply search for the point at which $F(x) = .5$.

Importance sampling is a very important variance reduction technique which can be used to great advantage in solving the rendering equation.

## 6. Application to the rendering equation

The monte carlo algorithms presented above can all be applied to a solution of the rendering algorithm. For example, sequential uniform sampling is used to sample the aperture for depth of field blur. Adaptive hierarchical integration is used to subsample the pixel. Importance sampling by splitting along the medians is used in choosing a direction to shoot the next ray. We store the lighting model as a summed area table [Crow 1985], giving a probability distribution function which can undergo binary search to find the median in a reflectance cell. Since we search for a median hyperplane of the lighting model, nonlinear transformations of the domain are not particularly important. We simply project the pair of input and output hemispheres onto the tangent plane.

It is interesting to compare the path solution to the conventional ray tracing algorithm. It is in fact quite easy to convert a conventional ray tracer to this algorithm. We essentially perform a conventional ray tracing algorithm, but instead of branching a tree of rays at each surface, we follow only one of the branches to give a path in the tree. We always shoot toward known light sources, which, of course, may be extended areas. Thus a schematic of ray tracing versus the integral equation method appears thus:




Now an important phenomenon is pointed out by this diagram. Due to the passivity of surfaces, it is widely known that the first generation rays as well as the light source rays are the most important to in terms of variance that they contribute to the pixel integral. Second and higher generation rays contribute much less to the variance. But conventional ray tracing expends the vast bulk of the work on precisely those rays which contribute least to the variance of the image, it shoots too many rays of higher generations. The integral equation method is not prone to this criticism. Because a path is a tree with branching ratio 1, there are as many different first generation rays as there are higher generation rays. This is very important for variance reduction for motion blur, depth of field, and other effects in distributed ray tracing.

This diagram also points out an alternative algorithm for conventional distributed ray tracing. Rather than shooting a branching tree, just shoot a path with the rays chosen probabilistically. For scenes with much reflection and refraction, this cuts down vastly on the number of ray object intersections to be computed for a given pixel and performs a remarkable speed up of ray tracing for very little programming work. However, for this new fast form of ray tracing—called *path tracing*—we have found that it is very important to maintain the correct proportion of reflection, refraction, and shadow ray types contributing to each pixel. Rather than choosing the ray type randomly, there are two alternatives. First, keep track of of the number of each type shot. Make sure the sample distribution of ray types closely matches the desired distribution by varying the the probability of each type so that it is more certain that the sample distribution matches. This is the approach we have actually implemented. A second approach is to let the ray types be chosen randomly but to scale the contribution of each ray type by the ratio of desired distribution to the resulting weighted sample distribution.

## 7. Results

Figures 5 and 6 show resulting images from the integral equation technique. At each surface element hit, a random variable was calculated from a distribution determined by the specular, diffuse, and transmission coefficients. This random variable was used to choose the shooting of one ray from the surface element. A random point was chosen on each light source to serve as a target for an illumination ray. The variance reduction methods actually used were multidimensional sequential sampling for choosing the diffuse direction, specular direction, and refracted direction of a new ray. Multidimensional sequential sampling was also used to choose points on the light sources and imaging aperture. Hierarchical integration was used for antialiasing the pixel values. No adaptive or nonuniform sampling was used for either of these images. It is clear that importance sampling would improve the variance of the image considerably. Although implementation of importance sampling is simple and straightforward it has not yet been done. Also, keeping track of the variance of each pixel and collecting sequential has shown to be a significant speed up. However, our program did not do this for these images, we shot a constant 40 paths per pixel.

Figure 5 shows a model rendered via two techniques. On the left side is the model rendered via the standard ray tracing technique (albeit with ambient coefficient set to 0 and the single branching ratio speedup mentioned above). The right image shows the result of rendering via the integral equation. Both images are 256 by 256 pixels with a fixed 40 paths per pixel. The images were computed on an IBM-4341. The first image took 401 minutes of CPU time, the second image took 533 minutes. Note that the area of the sphere in shadow is picking up ambient illumination missing in the ray tracing picture. Also light is bouncing off the bottom of the sphere and lighting up the base plane.

In figure 6 we show an image illustrating the power of the integral equation technique. All objects in the scene are a neutral grey except for the green glass balls and the base polygon (which is slightly reddish). Any color on the grey objects would be missing from a ray tracing image. Note that the green glass balls cast caustics on objects in the scene. There is color bleeding from the lightly colored base polygon onto the bottom of the oblate spheroid in the upper right. For simplicity and comparison purposes, the opaque surfaces in this scene are lambertian, but there is no restriction on the lighting models that can be used. Figure 6 is a 512 by 512 pixel image with 40 paths per pixel. It was computed on an IBM 3081 and consumed 1221 minutes of CPU time. Al Barr provided the model for this image.

ACKNOWLEDGMENTS Thanks to Al Barr, Tim Kay, RobCook, Jim Blinn, and the members of CS286 Computer Graphics Seminar, for technical discussions. I am grateful to IBM, Juan Rivero of the Los Angeles Science Center and Alan Norton of Yorktown Heights Research for donating large numbers of mainframe cycles to Caltech. I also wish to thank the reviewers for their many thoughtful comments.

### References

J.L. BENTLEY, J.H. FRIEDMAN "Data structures for range searching", ACM Comp. Surv., 11,4, pp.397-409., 1979.

S. CHANDRASEKAR *Radiative Transfer*, Oxford University Press, 1950.

M.F. COHEN, D.P. GREENBURG "The Hemi-cube: a Radiosity solution for complex environments", Computer Graphics 19,3, pp.31-40, 1985.

R.L. COOK, T. PORTER, L CARPENTER "Distributed ray tracing", Computer Graphics18,3, pp.137-146,1984.

R.L. COOK, "Stochastic sampling in computer graphics", to appear in ACM Transactions of Graphics

R. COURANT AND D. HILBERT, *Methods of mathematical physics* 2 vols., Interscience, New York 1953, 1962.

F.C. CROW "Summed area tables for texture mapping", Computer Graphics 18,3, pp.207-212, 1984.

R.P. FEYNMAN AND A.P. HIBBS *Quantum Mechanics and Path Integrals*, McGraw-Hill, New York 1965.

H. GOLDSTEIN *Classical Mechanics* Addison-Wesley, Reading, Mass. 1950.

C. M. GORAL, K.E. TORRANCE, D.P. GREENBURG "Modeling the interaction of light between diffuse surfaces", Computer Graphics 18,3, pp.213-222, 1984.

I.H. HALTON "A retrospective and prospective survey of the monte carlo method", SIAM Rev. 12, pp.1-63, 1970.

J.T. KAJIYA, B. VON HERZEN "Ray tracing volume densitites", Computer Graphics 18,3, pp.165-174, 1984.

M.E. LEE, R.A. REDNER, S.P. USELTON "Statistically Optimized Sampling for distributed ray tracing" Computer graphics v.19,3 pp.61-67.

T. NISHITA, E. NAKAMAE "Continuous tone representation of three dimensional objects taking account of shadows and interreflection", Computer Graphics19,3, pp.23-30, 1985.

R.Y. RUBENSTEIN *Simulation and the Monte Carlo Method*, J.Wiley, New York, 1981.

R. SIEGEL, J.R. HOWELL *Thermal Radiation Heat Transfer*, McGraw Hill, New York, 1981.

T. WHITTED "An improved illumination model for shaded display", Comm. ACM, 23,6, pp.343-349, June 1980.

S. YAKOWITZ, *et. al.* "Weighted monte carlo integration", SIAM J. Num. An. 15,6, pp.1289-1300, 1978.

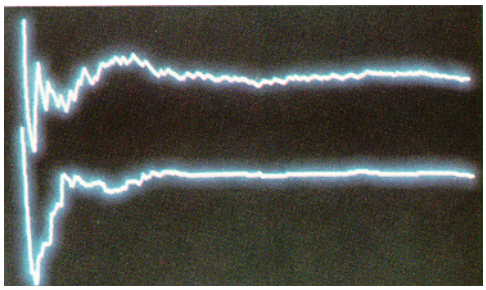


Figure 3. Convergence of naieve monte carlo vs. hierarchical integration. Shown are integral estimates as a function of number of samples cast. Naieve monte carlo is the top curve.

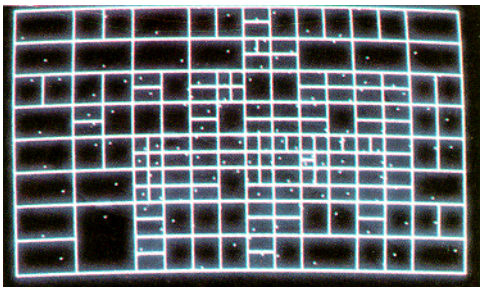


Figure 4. Subdivision of domain by adaptive hierarchical integration.

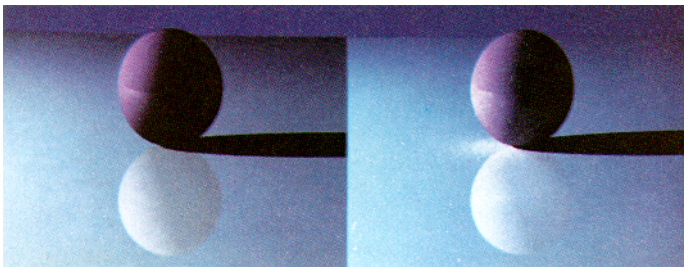


Figure 5. A comparison of ray tracing vs. integral equation technique. Note the presence of light on the base polygon scattered by the sphere from the light source.

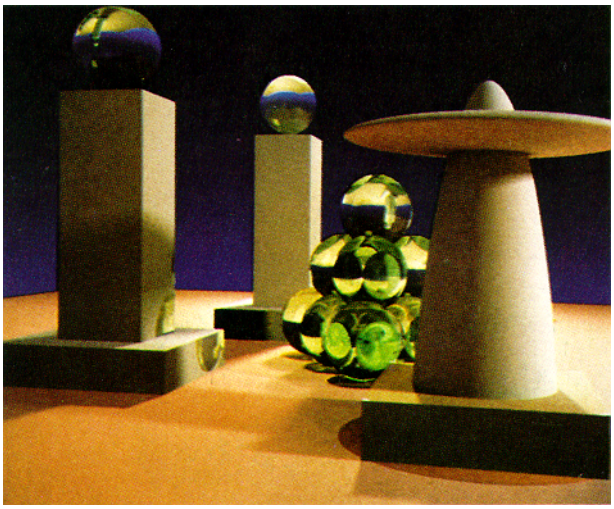


Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.