Metropolis Light Transport in Realistic Image Synthesis

Yunfeng Bai
mincomp@gmail.com

4/25/2011

1 Introduction

Computer graphics researchers had made great progresses towards realistic image synthesis in the past 30 years, from primitive Ray Tracing [6], to Avatar.

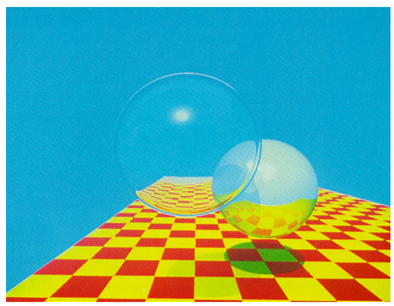

Figure 1: Primitive Ray Tracing

Figure 2: Avatar

The core mechanism behind realistic image synthesis is the simulation of light transportation. We can partition all the objects in real world into two categories: luminous objects and non-luminous objects. It's easier for us to get the colors of luminous objects; however, it's much harder in case of non-luminous objects, because non-luminous objects can absorb or reflect or refract light beams. So correct calculation of the radiance from non-luminous objects to eye is the main problem in realistic image synthesis. Following in this section are the basic physics concepts of light transport, which is heavily adopted on [1].

1.1 Basic Concepts

Radiant Power or Flux
Φ, called radiant power, or flux, is the energy flows through a surface per unit time.
Irradiance
E, called irradiance, is the incident radiant power on a surface, per unit surface area, defined as:

$$E = \frac{d\Phi}{dA}.$$

Radiance
L, called radiance, is the flux per unit projected area per unit solid angle. It expresses how much power goes through a certain point on a surface, per unit solid angle, and per unit projected area. Radiance is a function which varies with position x and direction vector Θ, so it's written as $L(x, \Theta)$. Finally, it's defined as:

$$L = \frac{d^2\Phi}{d\omega dA^{\perp}} = \frac{d^2\Phi}{d\omega dA\cos\theta},$$

where dw is the differential solid angle, $dA^{\perp}$ are the projected differential area and θ is the angle between the surface normal and Θ.
Relationships between flux, irradiance and radiance
According to previous definitions, the 3 quantities have the following relationships:

$$\Phi = \int_A \int_\Omega L(x \to \Theta) cos\theta d\omega_\Theta dA_x,$$

$$E(x) = \int_\Omega L(x \to \Theta) cos\theta d\omega_\Theta,$$

where A is the total surface area, Ω is the total solid angle at each point on the surface, $L(x \to \Theta)$ and $L(x \leftarrow \Theta)$ represents the radiance leaving/arriving point x from direction Θ, respectively.

1.2 BRDF

BRDF(bidirectional reflectance distribution function) is widely used in computer graphics, to describe the reflectance properties of surfaces. In BRDF, reflections on a surface are modeled as: at a given point x, for a given incident ray in direction Ψ that hits x, the reflected ray will start at x and in direction Θ, which carries a portion(which is denoted by BRDF) of the incident power at x. Formally, BRDF is defined as:

$$f_r(x, \Psi \to \Theta) = \frac{dL(x \to \Theta)}{dE(x \leftarrow \Psi)}.$$

BRDF can be acquired with empirical models or actual measurements, and we'll assume that we have known the function from now on.

1.3 The Rendering Equation

Now we have prepared all the concepts to build the rendering equation for describing light transportation. By conservation of energy, total outgoing radiance at a point x and some direction Θ is the sum of the emitted radiance and the reflected radiance at that point, if we denote the emitted radiance as $L_e(x \to \Theta)$ and the reflected radiance as $L_r(x \to \Theta)$, then

$$L(x \to \Theta) = L_e(x \to \Theta) + L_r(x \to \Theta).$$

Incorporating BRDF, the reflected radiance should be integrated through all possible incident directions Ψ:

$$L_r(x \to \Theta) = \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi,$$

where N_x is the surface normal at point x. In all,

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi.$$

Now we have a recursive integration equation, the rendering equation, to solve, and what we want to get is the radiance $L(x \to \bar{\Theta})$, where from point x through direction $\bar{\Theta}$ the outgoing radiance can arrive at the eye(or camera), so that we can accumulate the flux at each pixel.
Because rendering equation is in such a complex form, we need some special technique to calculate its value, which is Monte Carlo integration method introduced in the next section.

2 Monte Carlo Integration

Monte Carlo methods are a set of techniques that use statistical sampling to simulate phenomena or evaluate values of functions. The discussion of Monte Carlo integration in this section is based on the third chapter of [1].
Assume that we need to integrate a function: $\int_\Omega f(x)dx$, and we want to compute the integral through sampling $f(x)$. The samples are selected randomly over the integral domain Ω with probability distribution function $p(x)$. Now we define a function "estimator", I, as

$$I = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)},$$

where x_i's are i.i.d. random variables with distribution p.
The expectation of I, $E[I]$ is,

$$\begin{aligned} E[I] &= E[\frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}] \\ &= \frac{1}{N} E[\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}] \\ &= \frac{1}{N} N \int_\Omega \frac{f(x)}{p(x)} p(x) dx \\ &= \int_\Omega f(x) dx, \end{aligned}$$

actually the integral of $f(x)$.
With a few more calculations, the variance of the estimator is:

$$\sigma^2 = \frac{1}{N} \int (\frac{f(x)}{p(x)} - \int f(x)dx)^2 p(x) dx,$$

thus the variance decreases linearly as N increases. In other words, the error will be slowly reduced to 0.

2.1 The Path Formulation of Rendering Equation

Now let's explore the path formulation of rendering equation, so that we can use Monte Carlo integration technique to integrate it.
Expanding the rendering equation:

$$L(x \to \Theta) = L_e(x \to \Theta) + L_r(x \to \Theta),$$

noticing that $L_r(x \to \Theta)$ can be split into direct illumination(the radiance directly from luminous objects directly) and indirect illumination(the radiance received from non-luminous objects' reflections):

$$L_{direct} = \int_{\Omega_{direct}} f_r(x, \Psi \to \Theta) L_e(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$
$$L_{indirect} = \int_{\Omega_{indirect}} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$

and $\Omega_{direct} \cap \Omega_{indirect} = \Omega$, thus

$$\begin{aligned}
L_r(x \to \Theta) &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi = \\
&= L_{direct_x} + L_{indirect_x} \\
&= \int_{\Omega_{direct_x}} f_r(x, \Psi \to \Theta) L_e(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi + \int_{\Omega_{indirect_x}} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
&= \int_{\Omega_{direct_x}} f_r(x, \Psi \to \Theta) L_e(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi + \int_{\Omega_{indirect_x}} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
&= \int_{\Omega_{direct_x}} f_r(x, \Psi \to \Theta) L_e(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
&\quad + \int_{\Omega_{indirect_x}} f_r(x, \Psi \to \Theta) (L_{direct_y} + L_{indirect_y}) \cos(N_x, \Psi) d\omega_\Psi \\
&\quad \text{(y is first hit point of the ray from the x and in direction } \Psi\text{)} \\
&= \int_{\Omega_{direct_x}} f_r(x, \Psi \to \Theta) L_e(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
&\quad + \int_{\Omega_{indirect_x}} f_r(x, \Psi \to \Theta) (\int_{\Omega_{direct_y}} f_r(y, \Psi_y \to \Psi) L_e(y \leftarrow \Psi_y) \cos(N_y, \Psi_y) d\omega_{\Psi_y}) \cos(N_x, \Psi) d\omega_\Psi \\
&\quad + \int_{\Omega_{indirect_x}} f_r(x, \Psi \to \Theta) (\int_{\Omega_{indirect_y}} f_r(y, \Psi_y \to \Psi) L(y \leftarrow \Psi_y) \cos(N_y, \Psi_y) d\omega_{\Psi_y}) \cos(N_x, \Psi) d\omega_\Psi \\
&= \int_{\Omega_{direct_x}} f_r(x, \Psi \to \Theta) L_e(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \\
&\quad + \int_{\Omega_{indirect_x}} \int_{\Omega_{direct_y}} f_r(x, \Psi \to \Theta) f_r(y, \Psi_y \to \Psi) L_e(y \leftarrow \Psi_y) \cos(N_x, \Psi) \cos(N_y, \Psi_y) d\omega_\Psi d\omega_{\Psi_y} \\
&\quad + \int_{\Omega_{indirect_x}} \int_{\Omega_{indirect_y}} f_r(x, \Psi \to \Theta) f_r(y, \Psi_y \to \Psi) (L_{direct_z} + L_{indirect_z}) \cos(N_x, \Psi) \cos(N_y, \Psi_y) d\omega_\Psi d\omega_{\Psi_y}
\end{aligned} \tag{1}$$

if we continue expanding recursively, we can get the formulation:

$$L_r(x \to \Theta) = L_{zero\_reflection\_to\_x} + L_{one\_reflection\_to\_x} + L_{two\_reflection\_to\_x} + L_{three\_reflection\_to\_x} + ...$$

where $L_{zero\_reflection\_to\_x}$ is just L_{direct_x}, and

$$L_{one\_reflection\_to\_x} = \int_{\Omega_{indirect_x}} \int_{\Omega_{direct_y}} f_r(x, \Psi \to \Theta) f_r(y, \Psi_y \to \Psi) L_e(y \leftarrow \Psi_y) \cos(N_x, \Psi) \cos(N_y, \Psi_y) d\omega_\Psi d\omega_{\Psi_y}$$

is the radiance received through one reflection, similarly $L_{two\_reflection\_to\_x}$ is the radiance received through two reflections, and so on.

Now let's define Ω_k be the set of all paths of the form $\bar{x} = x_0 x_1 ... x_k$, where $k \geq 1$ and x_i's are surface points, and x_i and x_{i+1} can be observed by each other, then let's define

$d\mu(\bar{x}) = d\mu_k(x_0...x_k) = d\Psi_{x_0} d\Psi(x_1 \to x_0)...d\Psi(x_k \to x_{k-1})$. If we denote $\Omega^* = \bigcup \Omega_k$, we can write the rendering equation as:

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega^*} g(\bar{x}) d\mu(\bar{x}),$$

where $g(\bar{x}) = g(x_0...x_k) = L_e(x_k \to x_{k-1}) \cos(N_{x_0}, \Psi_{x_0}) \cos(N_{x_1}, \Psi(x_1 \to x_0))... \cos(N_{x_k}, \Psi(x_k \to x_{k-1}))$ So now we have this non-recursive path formulation of rendering equation. It's possible to use Monte Carlo integration by independently randomly sampling paths in Ω^* according to some distribution, and calculate $g(\bar{x})$.

Similar method is proposed in [2], which is now called the "Path Tracing Algorithm". Also we can reverse the whole procedure(intuitively, shoot light from the eye), and get similar equations. This method is called "Light Tracing". Later on in "Bidirectional Path Tracing" [3], researchers combined both path tracing and light tracing to produce better results.

2.2 Importance Sampling

In practice, we don't want to sample too many points, so we want the variance to be as small as possible (best if it's 0!). If we use Lagrangian multipliers to minimize the variance, we can get the optimal $p(x)$:

$$p(x) = \frac{|f(x)|}{\int f(x) dx},$$

in which case the variance is 0. However, $\int f(x)dx$ is what we want to solve, it's impossible to know its exact values beforehand.

Nevertheless, Metropolis sampling, which will be discussed in the next section, can magically sample points so that whose distribution converges to $\frac{|f(x)|}{\int f(x)dx}$.

3 Metropolis Light Transport

Metropolis Light Transport is proposed in [5], which adopted the method of Metropolis sampling [4] to the light transport problem. It's a robust algorithm that works well in variant scenes, and then became one of the standard tools for calculating global illumination.

3.1 Metropolis Sampling Algorithm

First let's give some formal definitions. We are given a state space Ω, a non-negative function $f{:}\omega \to \mathrm{R}^+$, and an initial state $\bar{X_0} \in \Omega$.

Our goal is to generate a random walk $\bar{X_0}$, $\bar{X_1}$, ..., so that $\bar{X_i}$ is eventually distributed to $p(\bar{X}) = \frac{f(\bar{X})}{\int f(\bar{X})d\bar{X}}$.

In the random walk each $\bar{X_i}$ is obtained from mutations of $\bar{X_{i-1}}$, thus this is a Markov chain.

The Metropolis sampling algorithm tries to construct a transition function K of $\bar{X_i}$, so that the stationary distribution of this Markov chain is $\frac{f(\bar{X})}{\int f(\bar{X})d\bar{X}}$.

Detailed Balance Condition

If we adopt the idea of DBC in this problem, then we'll get

$$\frac{f(\bar{X})}{\int f(\bar{X})d\bar{X}} K(\bar{X}, \bar{Y}) = \frac{f(\bar{Y})}{\int f(\bar{X})d\bar{X}} K(\bar{Y}, \bar{X})$$

multiply both sides by $\int f(\bar{X})d\bar{X}$, it becomes

$$f(\bar{X})K(\bar{X}, \bar{Y}) = f(\bar{Y})K(\bar{Y}, \bar{X})$$

then

$$\frac{K(\bar{X}, \bar{Y})}{K(\bar{Y}, \bar{X})} = \frac{f(\bar{Y})}{f(\bar{X})}$$

if we let $K(\bar{Y}, \bar{X}) = 1$, then $K(\bar{X}, \bar{Y}) = \frac{f(\bar{Y})}{f(\bar{X})}$.
So if we can construct a transition function K according to above equation, then we can be sure that $\frac{f(\bar{X})}{\int f(\bar{X}) d\bar{X}}$ is the stationary distribution.

The Acceptance Probability

Assume we already had a method to generate $\bar{X}_i$, with transition probability $T(\bar{X}, \bar{Y})$, we can let $K(\bar{X}, \bar{Y}) = T(\bar{X}, \bar{Y}) a(\bar{X}, \bar{Y})$, and $a(\bar{X}, \bar{Y}) = \frac{K(\bar{X}, \bar{Y})}{T(\bar{X}, \bar{Y})}$, called the acceptance probability. We accept each $\bar{Y}$ of mutation from X with probability $a(\bar{X}, \bar{Y})$ to take it into effect.

3.2 Metropolis Light Transport

Recall the path formulation of the rendering equation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega^*} g(\bar{x}) d\mu(\bar{x}),$$

we can solve this equation with path tracing, but carefully choosing the sampling distribution of $\bar{x}$ to reduce the variance.
With Metropolis sampling, now we can generate samples with distribution $\frac{g(\bar{x})}{\int g(\bar{x}) d\bar{x}}$ through random walk, thus the variance can be reduced effectively.

4 Discussion

4.1 The Rendering Equation

The rendering equation successfully grasps the essentials of most reflections/refractions in real world, from which we can generate images with impressive realism. However, there are several deficiencies in the rendering equation.

1. Deficiency of BRDF. The rendering equation uses BRDF, which is not able to model more complex surface properties, like subsurface scattering (the ray shots into the surface at x but comes out at y). Later researchers develop BRDF further to give the rendering equation the ability to describe the reflections on such surfaces.

2. Participating Media. In rendering equation, surfaces are generally considered as big and flat planes; however, in special cases like smoke or sand, where the surfaces are too small, it's hard for a ray to hit on such surfaces even with careful direction choices. Later researchers consider this kind of substances as "participating media" and developed rendering algorithms in such cases.

3. Human Perception. It's generally known that humans don't perceive radiance in linear scale, so rendering equation is not enough to describe what we see. Researchers also proposed algorithms in this direction to convert radiance to actually color values.

4.2 Metropolis Light Transport

Metropolis sampling greatly reduces the variance in Monte Carlo integration, thus accelerates the speed of image generation greatly.
Though in long run we can see $\bar{X}_i$'s as i.i.d. with distribution $\frac{g(\bar{x})}{\int g(\bar{x}) d\bar{x}}$, they are actually correlated because we generate them using random walk. So in the first few steps, we need some extra steps to reduce the bias. Another way to reduce this kind of bias is to run multiple random walks parallelly.
The other problem about Metropolis light transport is about the pre-chosen mutation T. Though in [5] the authors proposed some ad hoc transformation, there is no general mathematical guideline yet to choose T to speed up convergence.

References

[1] P. Dutre, K. Bala, and P. Bekaert. *Advanced global illumination*. AK Peters Ltd, 2006.

[2] J.T. Kajiya. The rendering equation. *ACM SIGGRAPH Computer Graphics*, 20(4):143–150, 1986.

[3] E.P. Lafortune and Y.D. Willems. Bi-directional path tracing. In *PROCEEDINGS OF THIRD INTERNATIONAL CONFERENCE ON COMPUTATIONAL GRAPHICS AND VISUALIZATION TECHNIQUES (COMPUGRAPHICS93*. Citeseer, 1993.

[4] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087, 1953.

[5] E. Veach and L.J. Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997.

[6] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.