CS294-13: Special Topics Lecture #3
Advanced Computer Graphics
University of California, Berkeley Handout Date ???

Global Illumination and the Rendering Equation

Lecture #3: Wednesday, 9 September 2009
Lecturer: Ravi Ramamoorthi
Scribe: Daniel Ritchie

1 Introduction to Global Illumination

Thus far in this course we have mainly considered *local illumination*; that is, light cast directly from light sources to surfaces. Needless to say, considering only local illumination leaves out a large number of visual effects that are critical for generating realistic imagery.

We now turn our attention to *global illumination*, characterized by multiple bounces of light between light sources and objects. This allows us to capture a variety of interesting effects. Some of them, such as shadows, reflections, and refraction, we have already seen in our discussion of ray tracing (Indeed, ray tracing is a form of global illumination algorithm, as we will see shortly). Others, such as caustics and color bleeding, are new (see Figure 1). We will need to develop a more sophisticated rendering framework in order to capture these effects.

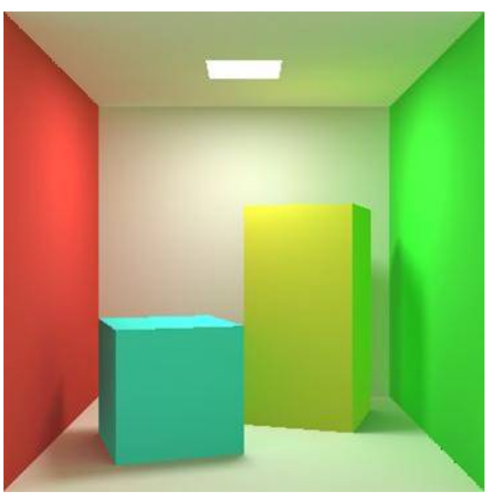

Figure 1: Color bleeding demonstrated in the Cornell Box (left); caustics produced by a wineglass (right).

Diffuse Inter-reflection/Color Bleeding: Diffuse inter-reflection refers to indirect light reflected off of diffuse surfaces onto nearby surfaces. In particular, if the diffuse surfaces have strong coloration, this leads to *color bleeding*, wherein the color of one surface appears to leak, or bleed, onto nearby surfaces. The Cornell box, originally designed by early global illumination researchers at Cornell University, is a classic testing environment for global illumination algorithms.

Caustics: When light focuses through a dielectric object, it produces a bright spot known as a *caustic*. An effect not reproducible by conventional radiosity methods, it has been the subject of much investigation in the late 1980's, 1990's, and even today.

2 Overview

In this lecture, we'll present a unified theory for all global illumination methods, including ray tracing, path tracing, and radiosity.

We'll first review the reflectance equation, then derive the Rendering Equation [Kajiya 86], a more complete description of light transport. This, it turns out, will give us a unified framework for all of our global illumination algorithms. We'll then discuss existing approaches to global illumination as special cases of Kajiya's Rendering Equation.

We'll give particular attention to deriving the equations for radiosity, a method for solving a simplified version of the Rendering Equation by assuming that all surfaces are perfect Lambertian (diffuse) reflectors. Path tracing, another popular global illumination algorithm, will be discussed in the next set of lectures.

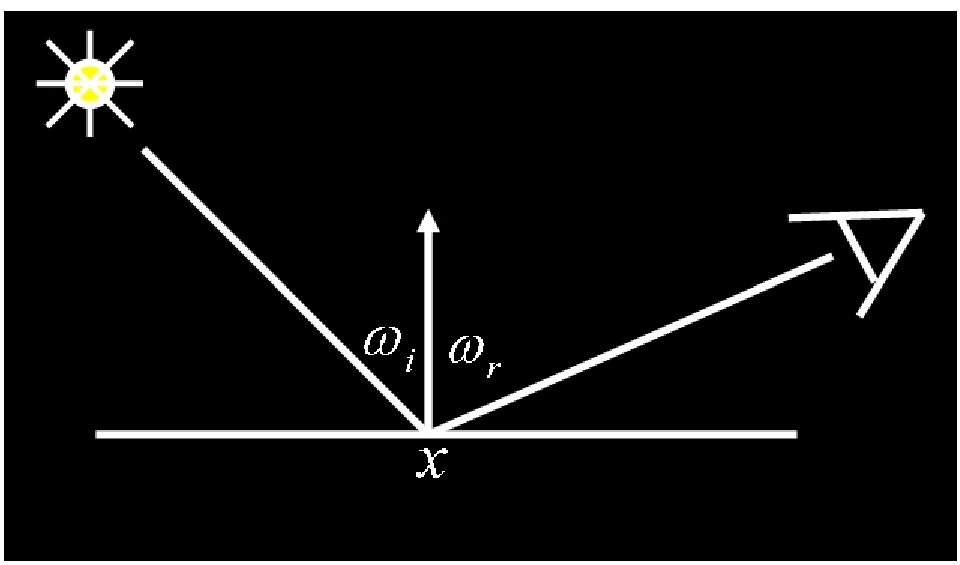

Figure 2: Local geometry for the reflectance equation.

3 Deriving the Rendering Equation

Recall the reflectance equation (Figure 2):

$$L_r(x, \omega_r) = L_e(x, \omega_r) + L_i(x, \omega_i) f(x, \omega_i, \omega_r)(\omega_i \cdot n)$$

This equation describes the reflected radiance L_r from a surface x in direction ω_r due to incident radiance L_i from a single light source. To bring this equation closer to a global description of light transport, we can first extend it to a sum over all light sources in the scene:

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \sum L_i(x, \omega_i) f(x, \omega_i, \omega_r)(\omega_i \cdot n)$$

This, however, is still an oversimplification. The incident irradiance on a surface is not all due to direct light sources; as we've seen, a significant portion of it comes from light reflected off of other nearby surfaces.

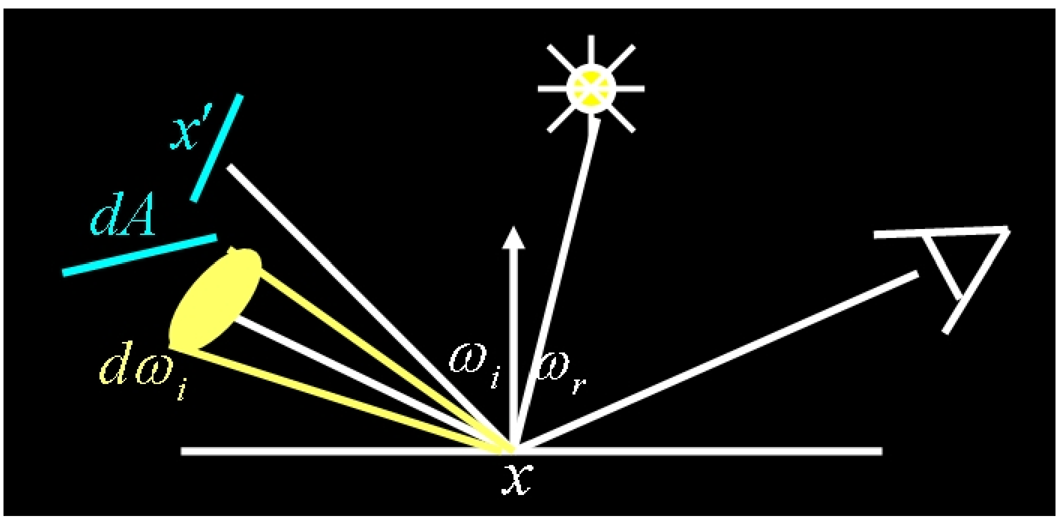

Figure 3: Local geometry for the rendering equation.

To capture this idea, we will generalize our equation. Instead of a sum over light sources, we can integrate over all solid angles in the visible hemisphere. We also replace the incident radiance L_i with the reflected radiance L_r from some other surface in the scene:

$$\begin{aligned} L_r(x, \omega_r) &= L_e(x, \omega_r) + \int_\Omega L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos\theta_i d\omega_i \\ &= L_e(x, \omega_r) + \int_\Omega L_r(x', -\omega_i) f(x, \omega_i, \omega_r) \cos\theta_i d\omega_i \end{aligned}$$

Light sources can be represented as emissive surfaces, so this equation is now fully general (see Figure 3).

This equation now globally describes the light transport in a scene, so we ought to be able to celebrate at this point. However, we've also gotten ourselves into a bind. In order to know the reflected radiance from a surface, we need to know the incident radiance from the other surfaces in the scene–but in order to know that, we need to know the reflected radiance from those surfaces, too! Put another way, "In order to know L_r, we must first know L_r."

In the equation above, L_e and f are known. The crux of the problem lies with the fact that the unknown quantity L_r is on the left-hand side of the equation *and* inside the integral. Fortunately for us, this type of problem has already been the subject of extensive study. In the mathematics community, it is known as a Fredholm Integral Equation of the second kind and has the following canonical form:

$$l(u) = e(u) + \int l(v)K(u, v)dv$$

where l is the unknown, e is known, and K is the kernel of the integral equation. For the inquisitive, more information is available here.

4 Solving the Rendering Equation

The Rendering Equation seems impenetrable as we've currently written it. Let's apply some linear operator theory to make things more manageable.

4.1 Linear Operator Theory

A linear operator acts on functions the way a matrix acts on vectors. In fact, one can think of a real-valued function as an infinite-dimensional vector where each "element" gives the value of the function when it is evaluated at a particular point.

$$h(u) = (M \circ f)(u)$$

Where M is a linear operator and f and h are functions of u.

Just like in the discrete linear algebra with which we're familiar, basic linearity rules still hold (they are called *linear* operators, after all):

$$M \circ (af + bg) = a(M \circ f) + b(M \circ g)$$

Many common operations on functions can be expressed as linear operators, including (and of interest to us) differentiation and integration:

$$(K \circ f)(u) = \int k(u, v)f(v)dv$$

$$(D \circ f)(u) = \frac{df}{du}(u)$$

We also define the idenity operator I, analagous to the identity matrix in the discrete case, to be the operator that takes every function to itself. In other words, $I \circ f = f$ for all functions f.

4.2 Solving in Linear Operator Form

With this new knowledge at our disposal, we can rewrite the Rendering Equation in linear operator form:

$$L = E + KL$$

Here, K is the operator representing integration against the kernel (which in our case is the BRDF f modulated by a cosine factor). E and L represent L_e and L_r–not the functions evaluated at a particular location and direction, but the functions themselves. If discretized, E becomes the vector of known light sources, L is the unknown reflected radiance at each measured surface point and outgoing angle, and K (which characterizes the reflectance of light around the scene) is typically called the *light transport matrix*.

With this notational simplification in place, it becomes straightforward to solve the Rendering Equation:

$$\begin{aligned}
L &= E + KL \\
IL - KL &= E \\
(I - K)L &= E \\
L &= (I - K)^{-1}E \\
&= (I + K + K^2 + K^3 + \ldots)E \\
&= E + KE + K^2E + K^3E + \ldots
\end{aligned}$$

The last two steps invoke the binomial series expansion of $(I - K)^{-1}$.

This result, in addition to its simplicity, has an amazingly intuitive physical explanation. The first term, E, gives the emission directly from the light sources. The next term, KE, describes the direct illumination on surfaces. The following term, K^2E, represents one-bounce indirect lighting. This sequence continues indefinitely, with the $(n-2)^{\text{nd}}$ term representing n-bounce indirect lighting. If we sum up infinitely many terms in this sequence, the sum converges to the exact solution of the Rendering Equation (see Figure 4).

4.3 Practical Approaches

Solving the Rendering Equation analytically proves intractably difficult. Instead, approximation solutions are used in practice.

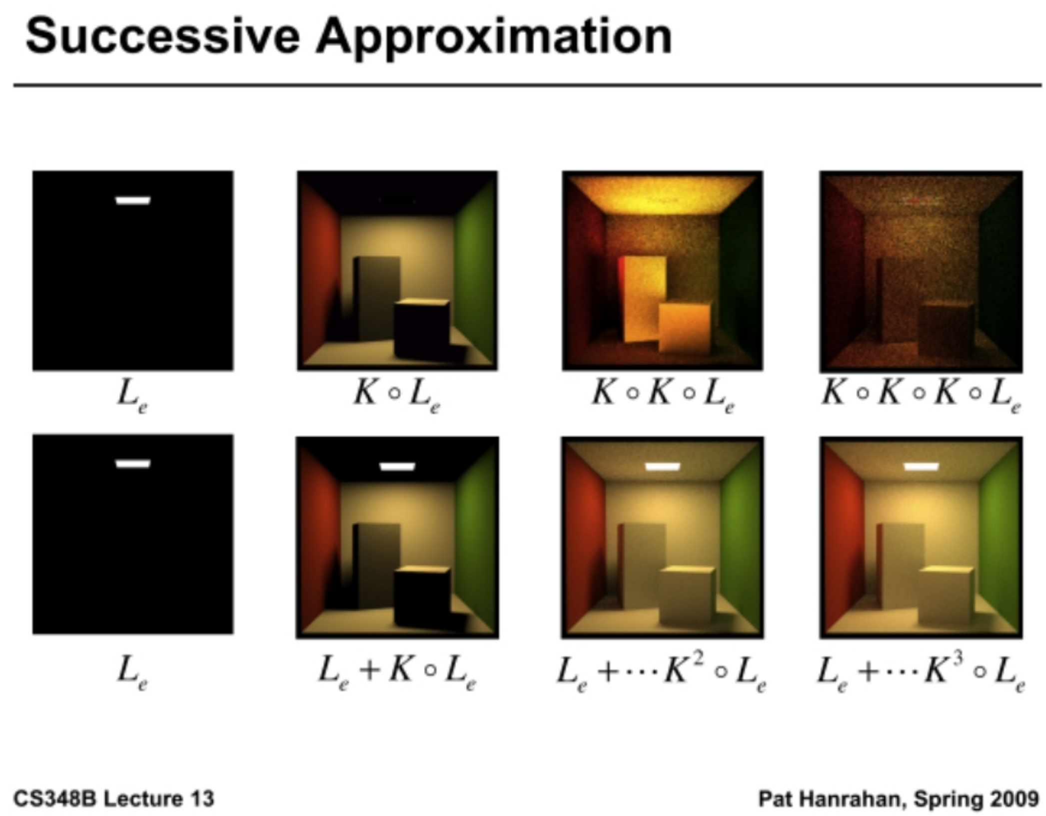

Figure 4: Accumulating successive bounces of light.

Finite element methods and Monte Carlo methods are the two most popular categories of algorithms used to solve the Rendering Equation. Finite element methods employ some form of discretization to reduce the Rendering Equation to a matrix equation. By contrast, Monte Carlo methods sample possible light paths, generating a statistical estimate of the appearance of a scene. Radiosity is a commonly-used incarnation of the finite element method, and ray/path tracing is a popular Monte Carlo approach.

Radiosity initially garnered much attention and excitement in the 1980's and early 90's with its ability to render complex diffuse inter-reflection. However, radiosity methods struggle with capturing complex (angle-dependent) reflectance as the scene must be discretized in greater than three dimensions. This leads to difficult meshing problems. As a result, Monte Carlo methods have surpassed radiosity and other finite element methods in popularity. However, radiosity is still frequently used in fields like architecture, where the scenes to be rendered typically behave in a mostly diffuse manner, and the ability to compute illumination once and then render interactive fly-throughs is highly desirable.

5 The (Final) Rendering Equation

As we have thus far derived it, the Rendering Equation has this form:

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_r(x', -\omega_i) f(x, \omega_i, \omega_r) \cos\theta_i d\omega_i$$

In a practical setting, integrating over the visible hemisphere is sometimes cumbersome or insufficient. It's much more convenient to integrate over all visible surfaces in the scene. To make this change of variables, we must first derive the equivalent form of $d\omega_i$:

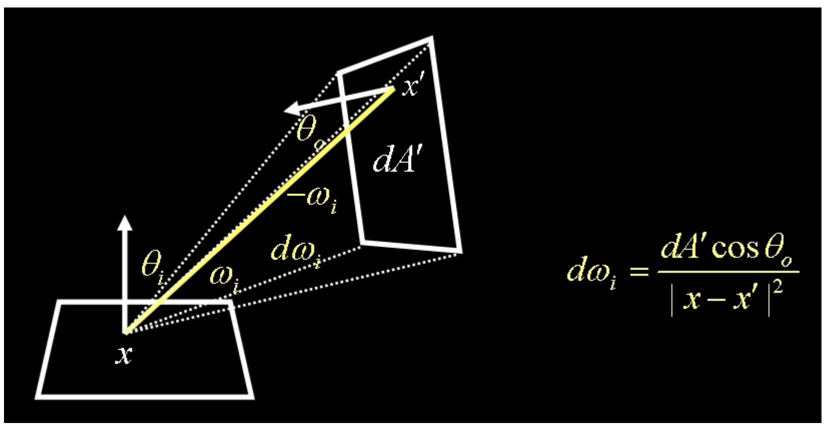

Figure 5: Deriving the surface parameterization of the rendering equation.

Now the Rendering Equation looks like this:

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\text{all visible } x'} L_r(x', -\omega_i) f(x, \omega_i, \omega_r) \frac{\cos\theta_i \cos\theta_o}{\|x - x'\|^2} dA'$$

This is still cumbersome, as the domain of the integral has become somewhat awkward. We can make it an integral over all surfaces in the scene if we introduce a binary visibility function V:

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\text{all } x'} L_r(x', -\omega_i) f(x, \omega_i, \omega_r) G(x, x') V(x, x') dA'$$

$$G(x, x') = G(x', x) = \frac{\cos\theta_i \cos\theta_o}{\|x - x'\|^2}$$

In this last step, we've also packaged up some terms into G, which we call the *geometry factor*. We now have, modulo minor notational differences, the canonical form of the Rendering Equation.

5.1 The Radiosity Equation

Without much extra work, we can also derive the discretized version of the Rendering Equation used by the radiosity method.

Given radiosity's assumption that all surfaces are perfectly diffuse, we can first drop the angular dependence of all terms in our equation:

$$L_r(x) = L_e(x) + f(x) \int_S L_r(x')G(x,x')V(x,x')dA'$$

Note that this also allows us to move the BRDF f outside the integral entirely.

Next, by convention, we change variables from radiance and BRDF to radiosity and albedo:

$$B(x) = E(x) + \rho(x) \int_S \frac{B(x')G(x,x')V(x,x')}{\pi} dA'$$

This equation concisely expresses the conservation of light energy at all points in space.

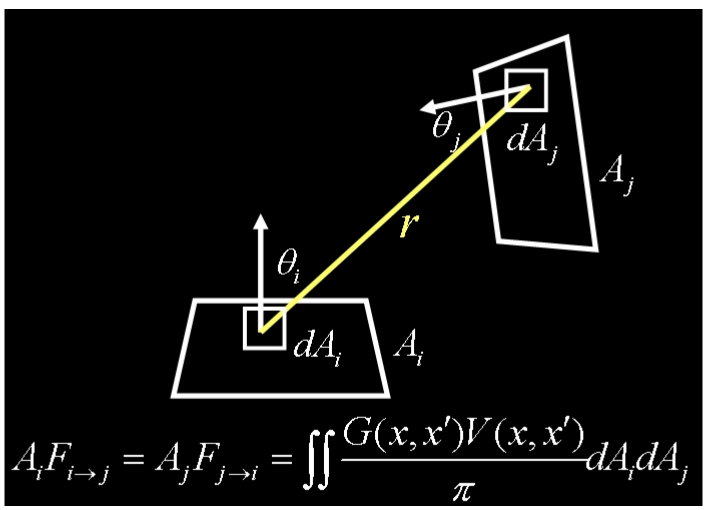

Figure 6: Deriving form factors for the radiosity equation.

We now take the plunge and discretize this equation over surfaces i,j in the scene:

$$B_i = E_i + \rho_i \sum_j B_j F_{j\to i}$$

The newly introduced term $F_{j\to i}$ is the *form factor* between i and j. Simply put, it is the fraction of light energy leaving surface patch j that arrives anywhere on patch i (see Figure 6). Methods exist to analytically compute form factors between two general polygons [Schroder & Hanrahan 93], but it is often necessary to compute form factors via ray tracing or hemicube rasterization when the space between the polygons is occupied by occluders.

The final step is to formulate this discretized version of the Rendering Equation as a matrix equation:

$$B_i = E_i + \rho_i \sum_j B_j F_{j \to i}$$

$$B_i - \rho_i \sum_j B_j F_{j \to i} = E_i$$

$$\sum_j M_{ij} B_j = E_i \qquad M_{ij} = I_{ij} - \rho_i F_{i \to j}$$

$$MB = E$$

where M and E are known and we must solve for B.

6 Conclusion

We've derived the Rendering Equation, a major theoretical development in physically-based rendering and the basis for a unifying framework for all global illumination algorithms. We've also seen how existing global illumination methods such as radiosity and ray tracing can be described as solving special cases of the problem posed by this equation.

In the next set of lectures, we'll explore the practical issues of numerically solving the Rendering Equation with Monte Carlo path tracing methods.