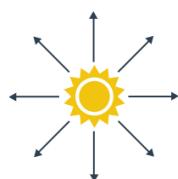


Illumination

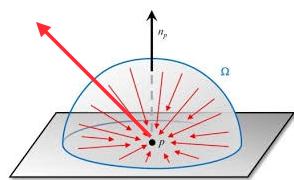
Luiz Velho
IMPA

Basic Concepts

- Light Sources



- Surfaces / Materials



Illumination Models

- Principle
 - Conservation of Energy
- Electromagnetic Model

$$\begin{aligned}\Phi_o &= F_r \Phi_i + F_t \Phi_i \\ F_r + F_t &= 1\end{aligned}$$

- Thermodynamic Model

$$\begin{aligned}\Phi_{tot} &= \bar{\mathbf{E}}_{tot} \\ \Phi_{out} &= \bar{\mathbf{E}} + \Phi_{in}\end{aligned}$$

The Rendering Equation

$$\phi(s, \omega) = E(s, \omega) + \int_s k(s, \omega', \omega) \phi(s, \omega') d\omega'$$

- **Kernel of the integral**

- Geometry
- Visibility
- Reflectivity

Radiant Energy

- **Flux of R.E.**

Phase Space

$$\phi(s, \omega) \quad \begin{array}{c} w \\ \nearrow \\ s \end{array} \quad \mathbb{R}^3 \times S^2$$

$$\phi = \frac{dQ}{dt}$$

Radiant energy passing through unity
of volume per unity of time

Transport of Energy

- **System in Equilibrium**

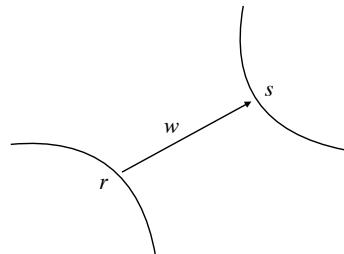
$$\frac{\partial \phi}{\partial t} = 0 \quad \text{i.e.,} \quad \phi = \text{constant}$$

Transport Equation

- Non-participating Medium (*vacuum*)

$$\phi(r, \omega) = \phi(s, \omega)$$

$$r, s \in \cup M_i \text{ and } \omega \in S^2$$

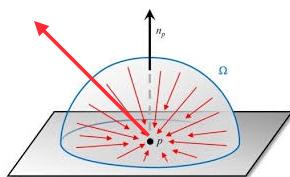


- Visibility Function btw Surfaces

$$\nu(r, \omega) \equiv \inf\{\alpha > 0 : (r - \alpha\omega) \in M\}$$

$$s = r - \nu(r, \omega)\omega$$

Illumination Hemisphere



- Equilibrium of Energy

$$\phi_o - \phi_i = \phi_e - \phi_a$$

outgoing incoming emitted absorbed

Boundary Conditions

- Explicit

$$\phi(s, \omega) = \mathcal{E}(s, \omega) \quad s \in \text{Lights}$$

- Implicit

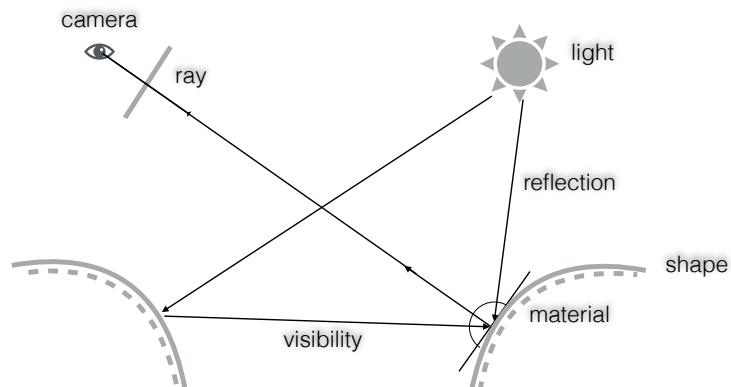
$$\begin{aligned} \phi(s, \omega) &= f_s(\phi(s, \omega')) && \text{scattering function} \\ &\uparrow_{out} && \uparrow_{in} \\ \phi(s, \omega) &= \int_{\Theta_i} k(s, \omega' \mapsto \omega) \phi(s, \omega') d\omega' \end{aligned}$$

OBS: Physical Limitation

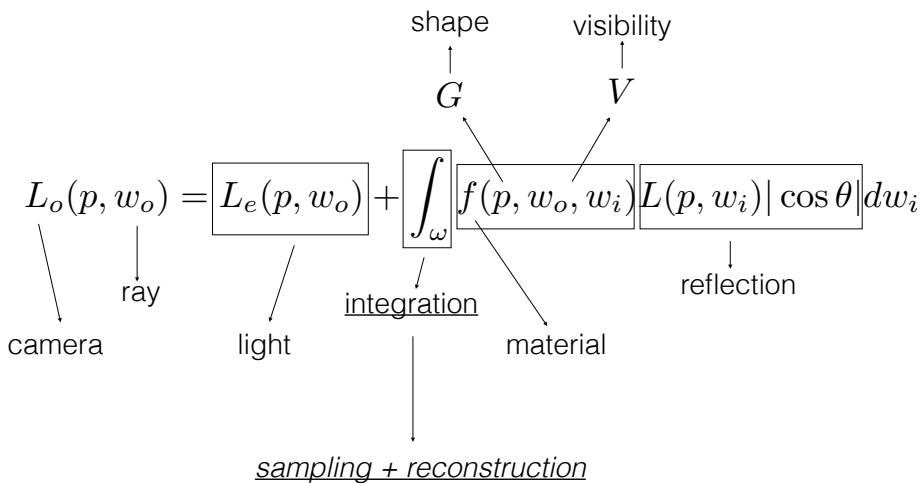
$$\int k d\omega \leq 1 \quad \text{and} \quad k \geq 0$$

In Practice

- Rendering Context



Anatomy of the Equation



Numerical Solution

★ Approximation

- Operator Notation

$$(Kf)(x) = \int k(x, y) f(y) dy \quad \text{Fredholm Eq. of 1st Kind}$$

- Radiance Equation in operator Form

$$L(r, \omega) = L^e(s, \omega) + (KL)(s, \omega)$$

or

$$L = L^e + KL$$

Method of Substitution

$$\begin{aligned} L &= L^e + K(L^e + KL) \\ &= L^e + KL^e + K^2L \end{aligned}$$

- repeating

$$\begin{aligned} &= L^e + KL^e + \dots + K^{n-1}L^e + K^nL \\ &= \lim_{n \rightarrow \infty} \sum K^n L^e \\ &\approx \sum_{i=0}^{n-1} K^i L^e \end{aligned}$$

Intuition: *Bounces of Light*

Approximating the Illumination Integral

$$I = g\mathbf{E} + g\mathbf{M}\mathbf{I}$$

$$\mathbf{M} = \int k(s, \omega', \omega)$$

- Newman Series

$$\begin{aligned} (1 - g\mathbf{M})I &= g\mathbf{E} \\ I &= (1 - g\mathbf{M})^{-1}g\mathbf{E} \\ I &= g\mathbf{E} + g\mathbf{M}g\mathbf{E} + g(\mathbf{M}g)^2\mathbf{E} + \dots \end{aligned}$$

Quality of Approximation

- **Error Analysis**

- Norm of $\| K \| < 1$

- **Residual**

$$e_n = \| M_\infty - M_n \|$$

- **Physical Interpretation**

- Direct Lighting (Local)
- Direct + Indirect Lighting (Global)

Computational Methods

- **Explicit Approximation** (Radiosity / Radiance)

- Compute L on Surfaces
- Sample L in Image
- *Finite Element Methods*
- Viewer Independent

- **Implicit Sampling** (Ray Tracing)

- Sample L in Image
- *Monte Carlo Methods*
- Viewer Dependent

Direct Lighting

Utah Solution

$$I = g\mathbf{E} + g\mathbf{M}\mathbf{E}_0$$

- Local Illumination
(only for light sources)
- No Shadows
- Direct Computation
- No Integration

Indirect Diffuse

$$\phi(s, \omega) = \mathbf{E}(s, \omega) + \int_S \mathbf{k}(s, \omega', \omega) \phi(r, \omega') d\omega'$$

Kernel of Integral

- Geometry
- Visibility
- Reflectivity

Diffuse

$$k(s, \omega', \omega) = \rho(s) \frac{\cos \theta_a \cos \theta_b}{\pi r_{ab}^2} V_{ab} = \rho(s) F_{a,b} V_{a,b}$$

Visibility

Form Factor

$$\phi(s, w) = E(s, w) + \rho(s) \int F_{a,b} V_{a,b} \phi(r, w') dw'$$

Radiosity

$$\mathbf{E} = (\mathbf{I} - \mathbf{gM})\mathbf{I} = \mathbf{GI}$$

- Global Illumination
(diffuse)
- View Independent

Finite Element Solution

$$\begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix} = \begin{pmatrix} 1 - \rho_1 F_{11} & \dots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & \dots & -\rho_2 F_{2n} \\ \vdots & & \vdots \\ -\rho_n F_{n1} & \dots & 1 - \rho_n F_{nn} \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{pmatrix}$$

Relaxation Computation

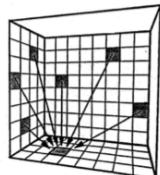
$$I_i^{(k)} = \mathbf{E} + \rho_i \sum_j \mathbf{F}_{ij} I_j^{(k-1)}$$

Radiosity Solution

• Methods

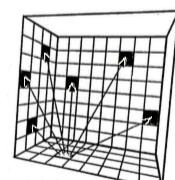
- Invert \mathbf{G} = $O(n^3)$ *Direct*
- Iterative = $O(n^2)$ *Gathering*
- Progressive < $O(n^2)$ *Shooting*

Gathering



Row of F times B
Calculate one row of F and discard

Shooting



Brightness order
Column of F times B

Ray Tracing

$$I = g\mathbf{E} + \mathbf{gM}_0\mathbf{g}\mathbf{E}_0 + \mathbf{g}(\mathbf{M}_0\mathbf{g})^2\mathbf{E}_0 + \dots$$

- Global Illumination
(specular)
- View Dependent

Stochastic Integration

- Monte Carlo Methods

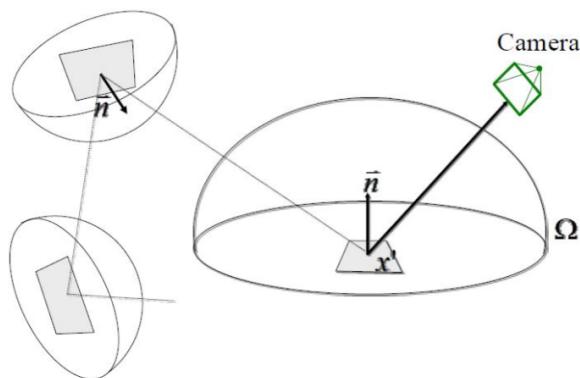
Recursive Computation

- Path Tracing
- Photon Mapping

Path Tracing

- Random Sampling

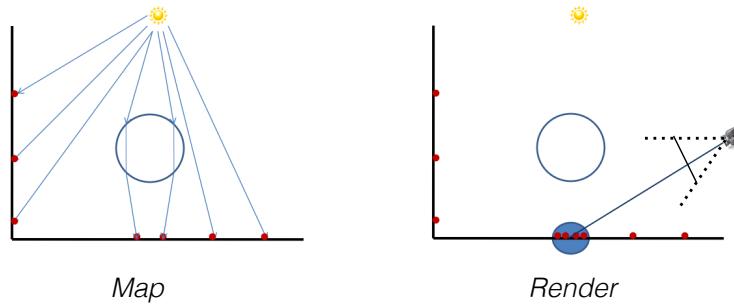
Estimate integral for each pixel by sampling paths from the camera



Photon Mapping

- **Two Pass Method**

1. Build Photon Map by tracing random paths from Lights
2. Render Image by tracing random paths from Camera



Global Illumination Algorithms

A Bit of History

- From Ideal to Physics Based Illumination

Graphics and Image Processing J.D. Foley
Editor

An Improved Illumination Model for Shaded Display

Turner Whitted
Bell Laboratories
Holmdel, New Jersey

To accurately render a voluminous range of three-dimensional scenes, global illumination information that affects the intensity of each pixel must be stored in a hierarchical tree. This information is stored in a tree of shaded surfaces, one for each pixel of the display and to the light source. Light rays from the scene are encountered and from there to other surfaces and to the light source. The shading calculations are performed at each node of the tree for each pixel of the display and passes it to the shading algorithm at the next level of the tree. To determine the intensity of the light received by the viewer, the shading algorithm at the root of the tree must shade to accurately simulate true reflection, shadows, and refraction, as well as the effects of global illumination. True reflection is included as an integral part of the visibility calculation. Surface displacement is used to model partially transparent objects.

Key Words and Phrases: computer graphics, computer rendering, visible surface algorithms, shading, raster displays.

CR Category: R.2

Introduction

Since its beginning, shaded computer graphics has progressed through various evolution. Even the earliest visible surface algorithms included shades that simulated the effect of light hitting a surface and the resulting transparency [H]. The importance of illumination models is most easily demonstrated by the results produced by the first computer graphics systems [F].

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage and that copies are not sold in whole or in part. As noted in [R], this drawback does not affect the quality of the results produced by the algorithm, but it seriously hinders the quality of specular reflections. A method developed by Whitted [W] (2) partially overcomes this shortcoming by modeling environment mapping and mapping it onto a sphere of infinite radius. The sphere is divided into a grid of small squares, and the component

© 1986 ACM 0-89791-096-2/86/0500-0103 \$00.75

The role of the illumination model is to determine how much light is reflected by the viewer from a visible point on a surface as a function of light source direction, surface orientation, and the properties of the surface and surface properties. The shading calculations can be performed using either an idealized or a physics-based model. Although the exact nature of reflection from surfaces is not fully understood, it is known that when two light rays meet the surface [3], most shaders produce between them a diffused reflection. Unfortunately, these models are usually limited in scope, failing to account for such phenomena as refraction, transmission, while getting the overall setting in which the scene is located wrong. In addition, these models are based on local geometry that traditional visible surface algorithms cannot provide the necessary global data.

In a direct rendering system, the shading algorithm must use global information to calculate intensities. Thus, to support this kind of shading, a ray tracing surface surface algorithm is presented.

1. Conventional Models

The simplest visible surface algorithms use shades based on Lambert's cosine law. The intensity of the light reflected by a surface is proportional to the angle of the surface normal and the light source direction, simulating漫反射 (diffuse reflection). This is a reasonable approximation to a diffuse, matte surface. A more sophisticated model is the Phong shading model [L]. Using Phong's model, the shading equation would be given by

$$I = I_r + \sum_{j=1}^n (S_j L_j) + k_s \sum_{j=1}^n (R_j L_j)^2 \quad (1)$$

where

I_r = the reflected intensity.
 L_j = the direction to the incident light.
 S_j = the diffuse reflection constant.
 R_j = the surface normal.
 k_s = the specular reflection coefficient.
 n = the number of light sources.
 L_j = the direction of the j th light source.
 S_j = the specular reflection coefficient.
 R_j = the direction of the j th light source.
 L_j = the direction of the j th light source.
 S_j = the specular reflection coefficient.
 R_j = the direction of the j th light source.

Phong's model assumes that each light source is treated independently, distant from the object in the scene. The model does not account for objects within a scene that may cast shadows on other objects, nor does it account for the effect of light scattering from other objects to object. As noted in [R], this drawback does not affect the quality of the results produced by the algorithm, but it seriously hinders the quality of specular reflections. A method developed by Whitted [W] (2) partially overcomes this shortcoming by modeling environment mapping and mapping it onto a sphere of infinite radius. The sphere is divided into a grid of small squares, and the component

© 1986 ACM 0-89791-096-2/86/0500-0103 \$00.75

Ray Tracing

Path Tracing

Whitted Ray Tracing

- Recursive Ray Tracing
 - Global Illumination of Ideal Materials (perfect diffuse & specular)
- Algorithm:
 - For each pixel, trace primary ray in direction \mathbf{V} to the first visible surface.
 - For each intersection trace secondary rays:
 - Shadow in direction \mathbf{L} to light sources
 - Reflected in direction \mathbf{R}
 - Refracted (transmitted) in direction \mathbf{T}
 - Calculate shading of pixel based on light attenuation

