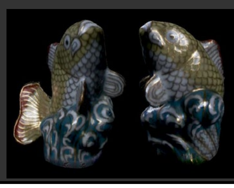
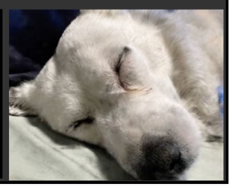
Image-Based Rendering

CSE 274, Lecture 2: Basics and Background

Ravi Ramamoorthi

http://www.cs.ucsd.edu/~ravir

1

Motivation: BRDFs, Ray Tracing, ...

- *Basics of Illumination, Reflection*
- *Formal radiometric analysis (not ad-hoc)*
- Ray Tracing
- Reflection Equation and Rendering Equation
- Monte Carlo Rendering next lecture

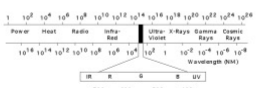
- Appreciate formal analysis in a graduate course, even if not absolutely essential in practice
- *Please e-mail re papers you want to present (by Th)*

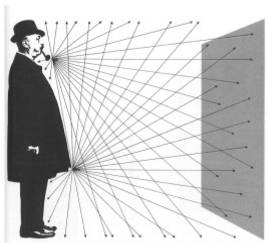
2

Light

Visible electromagnetic radiation

Power spectrum





From London and Upton

Polarization

Photon (quantum effects)

Wave (interference, diffraction)

CS348B Lecture 4 **Pat Hanrahan, 2009**

3

Radiometry

- Physical measurement of electromagnetic energy

- *Measure spatial (and angular) properties of light*
  - *Radiance, Irradiance*
  - Reflection functions: Bi-Directional Reflectance Distribution Function or BRDF
  - Reflection Equation
  - Simple BRDF models

4

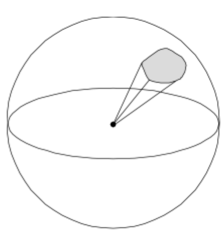
Angles and Solid Angles

- **Angle** $\theta = \frac{l}{r}$

  $\Rightarrow$ circle has 2π radians

- **Solid angle** $\Omega = \frac{A}{R^2}$

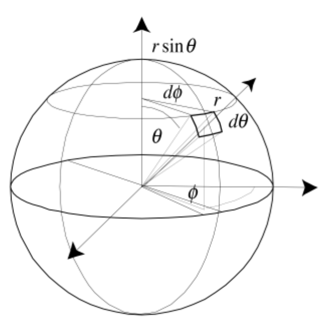
  $\Rightarrow$ sphere has 4π steradians



CS348B Lecture 4 **Pat Hanrahan, 2009**

5

Differential Solid Angles

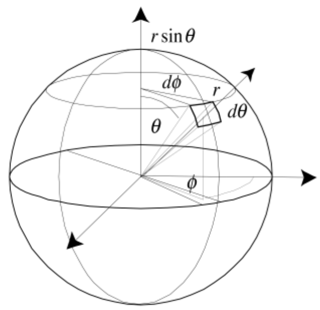


$$dA = (r\,d\theta)(r\sin\theta\,d\phi)$$
$$= r^2 \sin\theta\,d\theta\,d\phi$$

CS348B Lecture 4 **Pat Hanrahan, 2009**

6

Differential Solid Angles



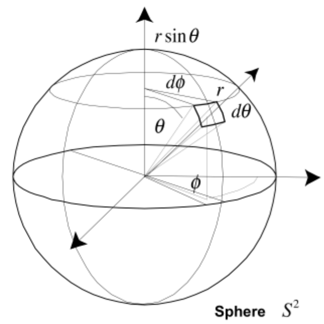
$$dA = (r\,d\theta)(r\sin\theta\,d\phi) = r^2 \sin\theta\,d\theta\,d\phi$$

$$d\omega = \frac{dA}{r^2} = \sin\theta\,d\theta\,d\phi$$

CS348B Lecture 4 — Pat Hanrahan, 2009

7

Differential Solid Angles



$$d\omega = \sin\theta\,d\theta\,d\phi$$

$$\Omega = \int_{S^2} d\omega = \int_0^{\pi}\int_0^{2\pi} \sin\theta\,d\theta\,d\phi = \int_{-1}^{1}\int_0^{2\pi} d\cos\theta\;d\phi = 4\pi$$

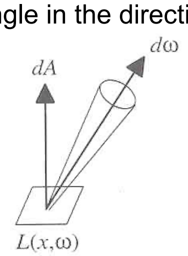
Sphere S^2

CS348B Lecture 4 — Pat Hanrahan, 2009

8

Radiance

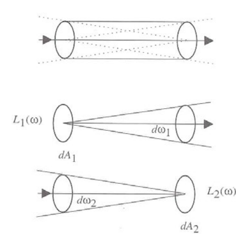
- Power per unit projected area perpendicular to the ray per unit solid angle in the direction of the ray
- Symbol: $L(x,\omega)$ (W/m^2 sr)
- Flux given by $d\Phi = L(x,\omega) \cos\theta \, d\omega \, dA$



9

Radiance properties

- Radiance constant as propagates along ray
  - Derived from conservation of flux
  - Fundamental in Light Transport.



$$d\Phi_1 = L_1 d\omega_1 dA_1 = L_2 d\omega_2 dA_2 = d\Phi_2$$

$$d\omega_1 = dA_2/r^2 \qquad d\omega_2 = dA_1/r^2$$

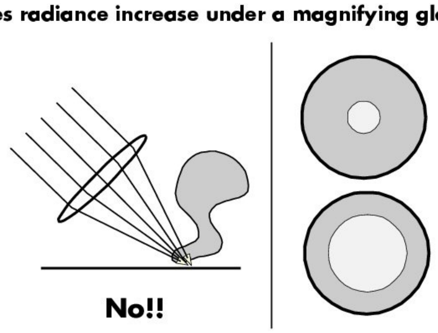
$$d\omega_1 dA_1 = \frac{dA_1 dA_2}{r^2} = d\omega_2 dA_2$$

$$\therefore L_1 = L_2$$

10

Quiz

Does radiance increase under a magnifying glass?



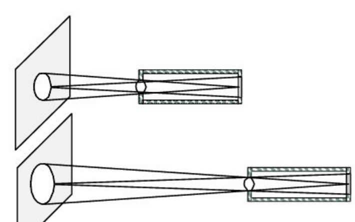
No!!

CS348B Lecture 4 — Pat Hanrahan, Spring 2002

11

Quiz

Does the brightness that a wall appears to the eye depend on the distance of the viewer to the wall?



No!!

CS348B Lecture 4 — Pat Hanrahan, Spring 2002

12

Radiance properties

- Sensor response proportional to radiance (constant of proportionality is throughput)
  - Far away surface: See more, but subtends smaller angle
  - Wall equally bright across viewing distances

Consequences

- Radiance associated with rays in a ray tracer
- Other radiometric quants derived from radiance

13

Irradiance, Radiosity

- Irradiance E is radiant power per unit area
- Integrate incoming radiance over hemisphere
  - Projected solid angle (cos θ dω)
  - Uniform illumination: Irradiance = π [CW 24,25]
  - Units: W/m^2
- Radiant Exitance (radiosity)
  - Power per unit area leaving surface (like irradiance)

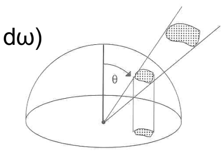


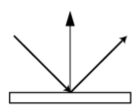
Figure 2.8: *Projection of differential area.*

14

Types of Reflection Functions

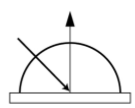
Ideal Specular

- **Reflection Law**
- **Mirror**



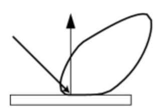
Ideal Diffuse

- **Lambert's Law**
- **Matte**



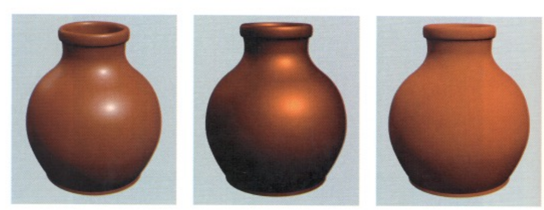
Specular

- **Glossy**
- **Directional diffuse**



15

Materials

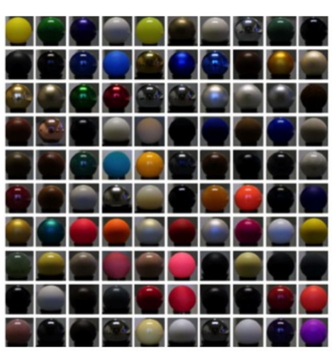


Plastic **Metal** **Matte**

From Apodaca and Gritz, *Advanced RenderMan*

16

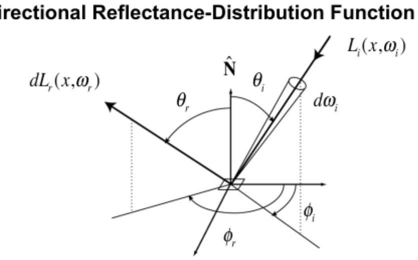
Spheres [Matusik et al.]



17

The BRDF

Bidirectional Reflectance-Distribution Function



$$f_r(\omega_i \to \omega_r) \equiv \frac{dL_r(\omega_i \to \omega_r)}{dE_i} \quad \left[\frac{1}{sr}\right]$$

18

Ideal Diffuse Reflection

Assume light is equally likely to be reflected in any output direction (independent of input direction).

$$L_{r,d}(\omega_r) = \int f_{r,d} L_i(\omega_i)\cos\theta_i \, d\omega_i$$
$$= f_{r,d} \int L_i(\omega_i)\cos\theta_i \, d\omega_i$$
$$= f_{r,d} E$$

$$M = \int L_r(\omega_r)\cos\theta_r \, d\omega_r = L_r \int \cos\theta_r \, d\omega_r = \pi L_r$$

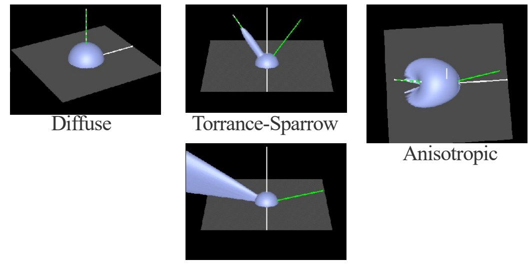
$$\rho_d = \frac{M}{E} = \frac{\pi L_r}{E} = \frac{\pi f_{r,d} E}{E} = \pi f_{r,d} \quad \Rightarrow \quad f_{r,d} = \frac{\rho_d}{\pi}$$

Lambert's Cosine Law $M = \rho_d E = \rho_d E_s \cos\theta_s$

CS348B Lecture 10 Pat Hanrahan, Spring 2002

19

Brdf Viewer plots



bv written by Szymon Rusinkiewicz

20

Torrance-Sparrow

Fresnel term: allows for wavelength dependency

Geometric Attenuation: reduces the output based on the amount of shadowing or masking that occurs.

$$f = \frac{F(\theta_i)G(\omega_i,\omega_r)D(\theta_h)}{4\cos(\theta_i)\cos(\theta_r)}$$

Distribution: distribution function determines what percentage of microfacets are oriented to reflect in the viewer direction.

How much of the macroscopic surface is visible to the light source

How much of the macroscopic surface is visible to the viewer

21

Motivation: BRDFs, Ray Tracing, ...

- Basics of Illumination, Reflection
- Formal radiometric analysis (not ad-hoc)
- *Ray Tracing*
- Reflection Equation and Rendering Equation
- Monte Carlo Rendering next lecture
- Appreciate formal analysis in a graduate course, even if not absolutely essential in practice
- *Please e-mail re papers you want to present (by Th)*

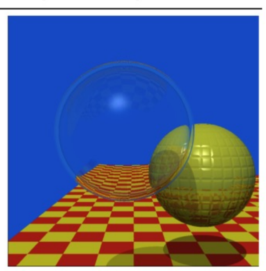
22

Ray Tracing History

Ray Tracing in Computer Graphics

"An improved Illumination model for shaded display," T. Whitted, CACM 1980

Resolution: 512 x 512
Time: VAX 11/780 (1979) 74 min. PC (2006) 6 sec.



Spheres and Checkerboard, T. Whitted, 1979

CS348B Lecture 2 Pat Hanrahan, Spring 2009

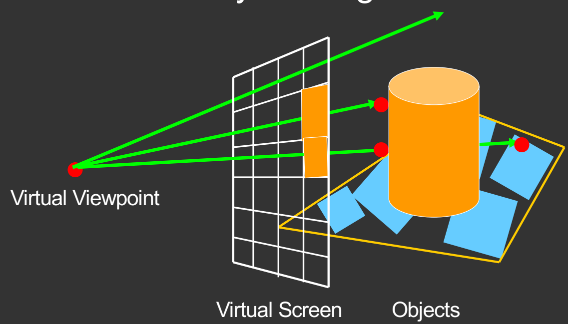
23

From SIGGRAPH 18

Real Photo: Instructor and Turner Whitted at SIGGRAPH 18

24

Ray Casting



25

Outline in Code

```
Image Raytrace (Camera cam, Scene scene, int width, int height)
{
   Image image = new Image (width, height) ;
   for (int i = 0 ; i < height ; i++)
        for (int j = 0 ; j < width ; j++) {
                Ray ray = RayThruPixel (cam, i, j) ;
                Intersection hit = Intersect (ray, scene) ;
                image[i][j] = FindColor (hit) ;
                }
   return image ;
}
```

26

Ray/Object Intersections

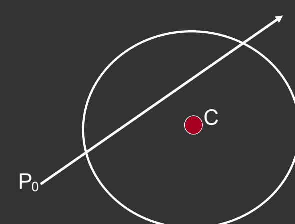
- Heart of Ray Tracer
  - One of the main initial research areas
  - Optimized routines for wide variety of primitives
- Various types of info
  - Shadow rays: Intersection/No Intersection
  - Primary rays: Point of intersection, material, normals
  - Texture coordinates
- Work out examples
  - Triangle, sphere, polygon, general implicit surface

27

Ray-Sphere Intersection

$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$sphere \equiv (\vec{P} - \vec{C}) \cdot (\vec{P} - \vec{C}) - r^2 = 0$$



28

Ray-Sphere Intersection

$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$sphere \equiv (\vec{P} - \vec{C}) \cdot (\vec{P} - \vec{C}) - r^2 = 0$$

Substitute

$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$sphere \equiv (\vec{P}_0 + \vec{P}_1 t - \vec{C}) \cdot (\vec{P}_0 + \vec{P}_1 t - \vec{C}) - r^2 = 0$$

Simplify

$$t^2(\vec{P}_1 \cdot \vec{P}_1) + 2t\,\vec{P}_1 \cdot (\vec{P}_0 - \vec{C}) + (\vec{P}_0 - \vec{C}) \cdot (\vec{P}_0 - \vec{C}) - r^2 = 0$$

29

Ray-Sphere Intersection

$$t^2(\vec{P}_1 \cdot \vec{P}_1) + 2t\,\vec{P}_1 \cdot (\vec{P}_0 - \vec{C}) + (\vec{P}_0 - \vec{C}) \cdot (\vec{P}_0 - \vec{C}) - r^2 = 0$$

Solve quadratic equations for t

- 2 real positive roots: pick smaller root
- Both roots same: tangent to sphere
- One positive, one negative root: ray origin inside sphere (pick + root)
- Complex roots: no intersection (check discriminant of equation first)

30

Ray-Sphere Intersection

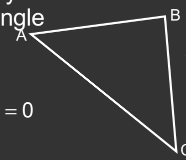
- Intersection point: $ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$
- Normal (for sphere, this is same as coordinates in sphere frame of reference, useful other tasks)

$$normal = \frac{\vec{P} - \vec{C}}{\left|\vec{P} - \vec{C}\right|}$$

31

Ray-Triangle Intersection

- One approach: Ray-Plane intersection, then check if inside triangle
- Plane equation:



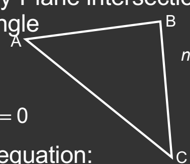
$$n = \frac{(C-A)\times(B-A)}{\left|(C-A)\times(B-A)\right|}$$

$$plane \equiv \vec{P} \bullet \vec{n} - \vec{A} \bullet \vec{n} = 0$$

32

Ray-Triangle Intersection

- One approach: Ray-Plane intersection, then check if inside triangle
- Plane equation:



$$n = \frac{(C-A)\times(B-A)}{\left|(C-A)\times(B-A)\right|}$$

$$plane \equiv \vec{P} \bullet \vec{n} - \vec{A} \bullet \vec{n} = 0$$

- Combine with ray equation:

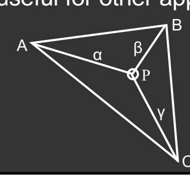
$$ray \equiv \vec{P} = \vec{P}_0 + \vec{P}_1 t$$

$$(\vec{P}_0 + \vec{P}_1 t) \bullet \vec{n} = \vec{A} \bullet \vec{n} \qquad t = \frac{\vec{A} \bullet \vec{n} - \vec{P}_0 \bullet \vec{n}}{\vec{P}_1 \bullet \vec{n}}$$

33

Ray inside Triangle

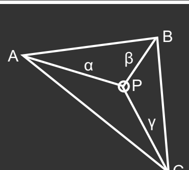
- Once intersect with plane, still need to find if in triangle
- Many possibilities for triangles, general polygons (point in polygon tests)
- We find parametrically [barycentric coordinates]. Also useful for other applications (texture mapping)



$$P = \alpha A + \beta B + \gamma C$$
$$\alpha \ge 0, \beta \ge 0, \gamma \ge 0$$
$$\alpha + \beta + \gamma = 1$$

34

Ray inside Triangle



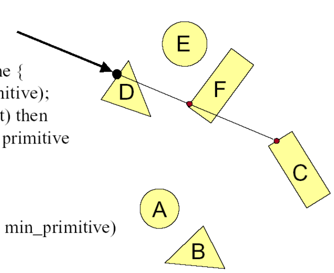
$$P = \alpha A + \beta B + \gamma C$$
$$\alpha \ge 0, \beta \ge 0, \gamma \ge 0$$
$$\alpha + \beta + \gamma = 1$$

$$P - A = \beta(B-A) + \gamma(C-A)$$
$$0 \le \beta \le 1 \ , \ 0 \le \gamma \le 1$$
$$\beta + \gamma \le 1$$

35

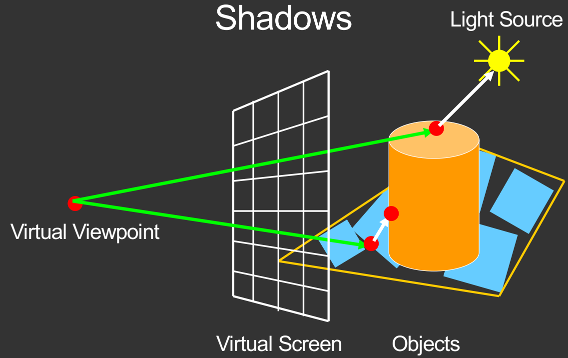
Ray Scene Intersection

```
Intersection FindIntersection(Ray ray, Scene scene)
{
    min_t = infinity
    min_primitive = NULL
    For each primitive in scene {
        t = Intersect(ray, primitive);
        if (t > 0 && t < min_t) then
            min_primitive = primitive
            min_t = t
        }
    }
    return Intersection(min_t, min_primitive)
}
```



36

Shadows

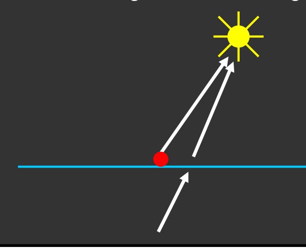


Shadow ray to light is blocked: object in shadow

37

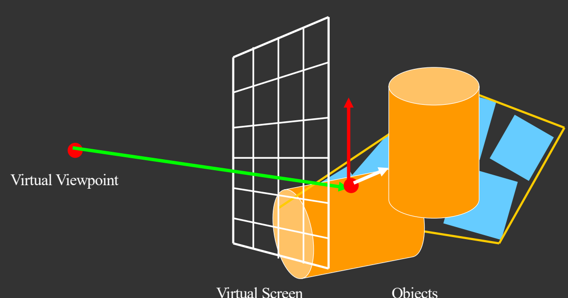
Shadows: Numerical Issues

- Numerical inaccuracy may cause intersection to be below surface (effect exaggerated in figure)
- Causing surface to incorrectly shadow itself
- Move a little towards light before shooting shadow ray



38

Mirror Reflections/Refractions



Generate reflected ray in mirror direction,
Get reflections and refractions of objects

39

Recursive Ray Tracing

For each pixel
- Trace Primary Eye Ray, find intersection
- Trace Secondary Shadow Ray(s) to all light(s)
  - Color = Visible ? Illumination Model : 0 ;
- Trace Reflected Ray
  - Color += reflectivity * Color of reflected ray

- Need acceleration structure for performance

40

Interactive Raytracing

- Ray tracing historically slow
- Now viable alternative for complex scenes
  - Key is sublinear complexity with acceleration; need not process all triangles in scene
- Allows many effects hard in hardware
- Today graphics hardware and software (NVIDIA Optix 6, RTX chips 10G+ rays per second). Video

41

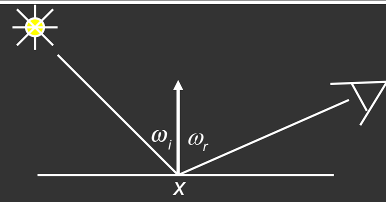
Motivation: BRDFs, Ray Tracing, ...

- Basics of Illumination, Reflection
- Formal radiometric analysis (not ad-hoc)
- Ray Tracing
- *Reflection Equation and Rendering Equation*
- Monte Carlo Rendering next lecture

- Appreciate formal analysis in a graduate course, even if not absolutely essential in practice
- *Please e-mail re papers you want to present (by Th)*

42

Reflection Equation

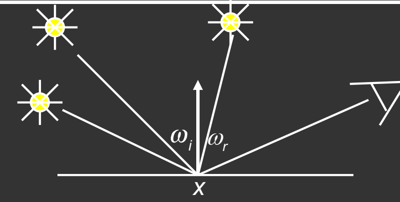


$$L_r(x,\omega_r) = L_e(x,\omega_r) + L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

43

Reflection Equation



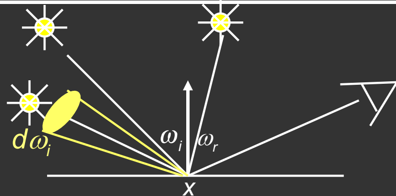
Sum over all light sources

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

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

44

Reflection Equation



Replace sum with integral

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

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

45

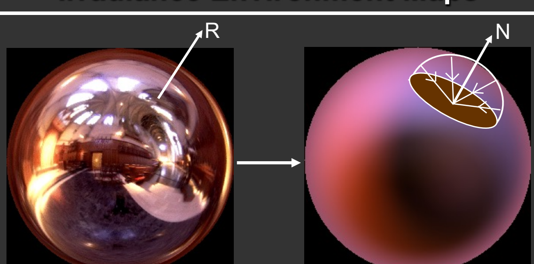
Environment Maps

- Light as a function of direction, from entire environment
- Captured by photographing a chrome steel or mirror sphere
- Accurate only for one point, but distant lighting same at other scene locations (typically use only one env. map)

Blinn and Newell 1976, Miller and Hoffman, 1984
Later, Greene 86, Cabral et al. 87

46

Irradiance Environment Maps



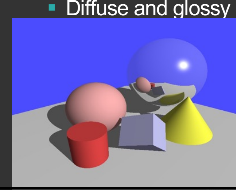
Incident Radiance (Illumination Environment Map) | Irradiance Environment Map

47

Illumination Models

Local Illumination
- Light directly from light sources to surface
- No shadows (cast shadows are a global effect)

Global Illumination: multiple bounces (indirect light)
- Hard and soft shadows
- Reflections/refractions (already seen in ray tracing)
- Diffuse and glossy interreflections (radiosity, caustics)



Some images courtesy Henrik Wann Jensen

48

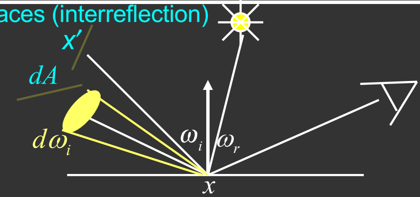
The Challenge

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_i(x,\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i \, d\omega_i$$

- Computing reflectance equation requires knowing the incoming radiance from surfaces
- But determining incoming radiance requires knowing the reflected radiance from surfaces

49

Rendering Equation



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

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

50

Outline

- Reflectance Equation
- Global Illumination
- Rendering Equation
- *As a general Integral Equation and Operator*
- *Approximations (Ray Tracing, Radiosity)*
- Surface Parameterization (Standard Form)

51

Rendering Equation (Kajiya 86)

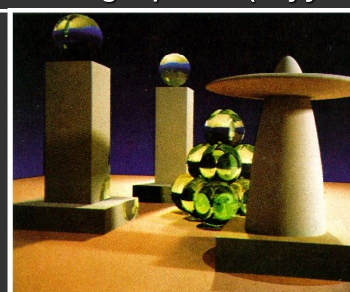


Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.

52

Rendering Equation as Integral Equation

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

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

Is a Fredholm Integral Equation of second kind [extensively studied numerically] with canonical form

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

Kernel of equation

53

Linear Operator Theory

- Linear operators act on functions like matrices act on vectors or discrete representations

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

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

- Basic linearity relations hold

a and b are scalars
f and g are functions

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

- Examples include integration and differentiation

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

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

54

Linear Operator Equation

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

Kernel of equation
Light Transport Operator

$$L = E + KL$$

Can be discretized to a simple matrix equation
[or system of simultaneous linear equations]
(L, E are vectors, K is the light transport matrix)

55

Solving the Rendering Equation

- Too hard for analytic solution, numerical methods
- Approximations, that compute different terms, accuracies of the rendering equation
- Two basic approaches are ray tracing, radiosity. More formally, Monte Carlo and Finite Element. Today Monte Carlo path tracing is core rendering method
- Monte Carlo techniques sample light paths, form statistical estimate (example, path tracing)
- Finite Element methods discretize to matrix equation

56

Solving the Rendering Equation

- General linear operator solution. Within raytracing:
- General class numerical ***Monte Carlo*** methods
- Approximate set of all paths of light in scene

$$L = E + KL$$
$$IL - KL = E$$
$$(I - K)L = E$$
$$L = (I - K)^{-1}E$$

Binomial Theorem

$$L = (I + K + K^2 + K^3 + ...)E$$
$$L = E + KE + K^2E + K^3E + ...$$

Term n corresponds to n bounces of light

57

Ray Tracing

$$L = E + KE + K^2E + K^3E + ...$$

- E: Emission directly From light sources
- KE: Direct Illumination on surfaces
- K^2E: Global Illumination (One bounce indirect) [Mirrors, Refraction]
- K^3E: (Two bounce indirect) [Caustics etc]

58

Ray Tracing

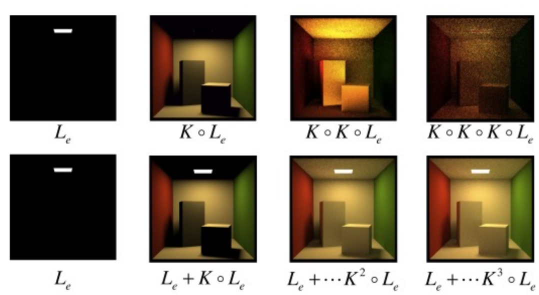
$$L = E + KE + K^2E + K^3E + ...$$

- E: Emission directly From light sources
- KE: Direct Illumination on surfaces
- K^2E: Global Illumination (One bounce indirect) [Mirrors, Refraction]
- K^3E: (Two bounce indirect) [Caustics etc]

OpenGL Shading

59

Successive Approximation



CS348B Lecture 13 Pat Hanrahan, Spring 2009

60

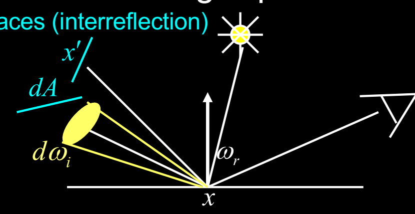
Outline

- Reflectance Equation (review)
- Global Illumination
- Rendering Equation
- As a general Integral Equation and Operator
- Approximations (Ray Tracing, Radiosity)
- *Surface Parameterization (Standard Form)*

61

Rendering Equation

Surfaces (interreflection)



$$\omega_i \sim x' - x$$

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

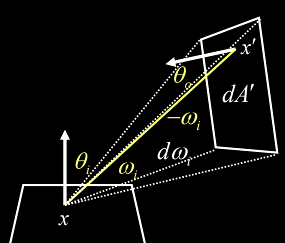
Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

62

Change of Variables

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

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)



$$d\omega_i = \frac{dA' \cos\theta_o}{|x - x'|^2}$$

63

Change of Variables

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

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

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

$$d\omega_i = \frac{dA' \cos\theta_o}{|x - x'|^2}$$

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

64

Rendering Equation: Standard Form

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

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

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

Domain integral awkward. Introduce binary visibility fn V

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

$$d\omega_i = \frac{dA' \cos\theta_o}{|x - x'|^2}$$

Same as equation 2.52 Cohen Wallace. It swaps primed And unprimed, omits angular args of BRDF, - sign.
Same as equation above 19.3 in Shirley, except he has no emission, slightly diff. notation

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

65