1

Illumination

Adam Finkelstein

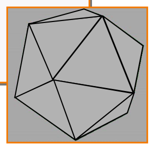
Princeton University

C0S 426, Fall 2001

2

Ray Casting

```
Image RayCast(Camera camera, Scene scene, int width, int height)
{
    Image image = new Image(width, height);
    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            Ray ray = ConstructRayThroughPixel(camera, i, j);
            Intersection hit = FindIntersection(ray, scene);
            image[i][j] = GetColor(scene, ray, hit);
        }
    }
    return image;
}
```

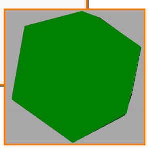


Wireframe

3

Ray Casting

```
Image RayCast(Camera camera, Scene scene, int width, int height)
{
    Image image = new Image(width, height);
    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            Ray ray = ConstructRayThroughPixel(camera, i, j);
            Intersection hit = FindIntersection(ray, scene);
            image[i][j] = GetColor(scene, ray, hit);
        }
    }
    return image;
}
```

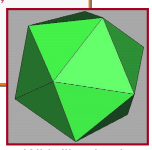


Without Illumination

4

Ray Casting

```
Image RayCast(Camera camera, Scene scene, int width, int height)
{
    Image image = new Image(width, height);
    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            Ray ray = ConstructRayThroughPixel(camera, i, j);
            Intersection hit = FindIntersection(ray, scene);
            image[i][j] = GetColor(scene, ray, hit);
        }
    }
    return image;
}
```



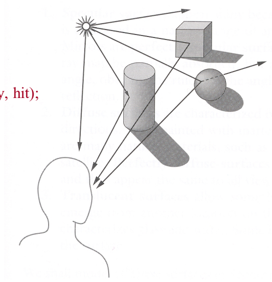
With Illumination

5

Illumination

- How do we compute radiance for a sample ray?

image[i][j] = GetColor(scene, ray, hit);

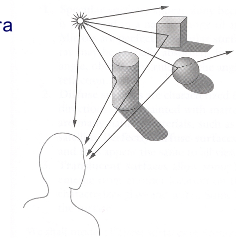


Angel Figure 6.2

6

Goal

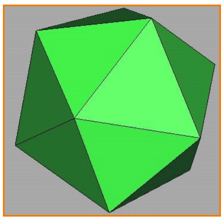
- Must derive computer models for ...
  - o Emission at light sources
  - o Scattering at surfaces
  - o Reception at the camera
- Desirable features …
  - o Concise
  - o Efficient to compute
  - o “Accurate”



7

Overview

- Direct Illumination
  - Emission at light sources
  - Scattering at surfaces
- Global illumination
  - Shadows
  - Refractions
  - Inter-object reflections

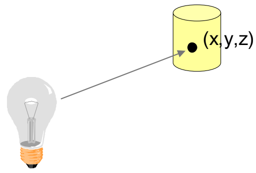


Direct Illumination

8

Modeling Light Sources

- $I_L(x,y,z,\theta,\phi,\lambda)$ ...
  - describes the intensity of energy,
  - leaving a light source, …
  - arriving at location(x,y,z), ...
  - from direction (θ,ϕ), ...
  - with wavelength λ

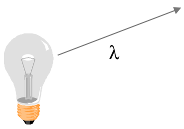


Light

9

Empirical Models

- Ideally measure irradiant energy for “all” situations
  - Too much storage
  - Difficult in practice



10

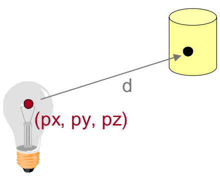
OpenGL Light Source Models

- Simple mathematical models:
  - Point light
  - Directional light
  - Spot light

11

Point Light Source 

- Models omni-directional point source
  - intensity (I_0),
  - position (px, py, pz),
  - factors (k_c, k_l, k_q) for attenuation with distance (d)



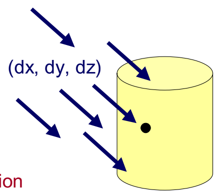
Light

$$I_L = \frac{I_0}{k_c + k_l d + k_q d^2}$$

12

Directional Light Source 

- Models point light source at infinity
  - intensity (I_0),
  - direction (dx,dy,dz)



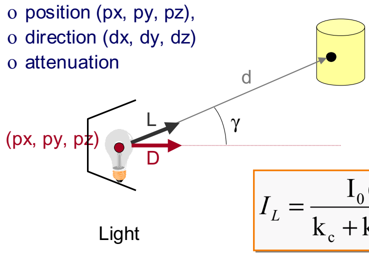
No attenuation
with distance

$$I_L = I_0$$

Spot Light Source

13

- Models point light source with direction
  - o intensity (I_0),
  - o position (px, py, pz),
  - o direction (dx, dy, dz)
  - o attenuation

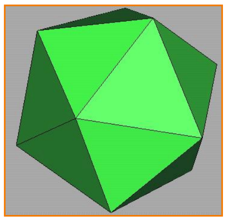


$$I_L = \frac{I_0 (D \bullet L)}{k_c + k_l d + k_q d^2}$$

Overview

14

- Direct Illumination
  - o Emission at light sources
  - o Scattering at surfaces
- Global illumination
  - o Shadows
  - o Refractions
  - o Inter-object reflections

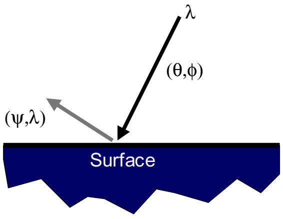


Direct Illumination

Modeling Surface Reflectance

15

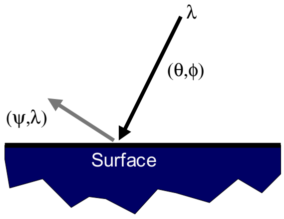
- $R_s(\theta,\phi,\gamma,\psi,\lambda)$ ...
  - o describes the amount of incident energy,
  - o arriving from direction (θ,ϕ), ...
  - o leaving in direction (γ,ψ), …
  - o with wavelength λ



Empirical Models

16

- Ideally measure radiant energy for “all” combinations of incident angles
  - o Too much storage
  - o Difficult in practice

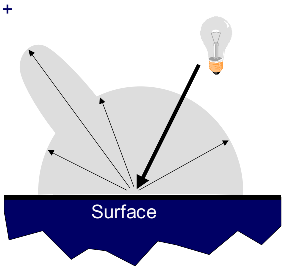


OpenGL Reflectance Model

17

- Simple analytic model:
  - o diffuse reflection +
  - o specular reflection +
  - o emission +
  - o “ambient”

Based on model proposed by Phong

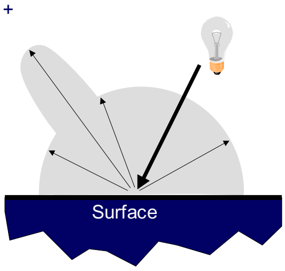


OpenGL Reflectance Model

18

- Simple analytic model:
  - o diffuse reflection +
  - o specular reflection +
  - o emission +
  - o “ambient”

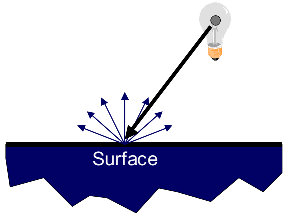
Based on model proposed by Phong



19

Diffuse Reflection

- Assume surface reflects equally in all directions
  - Examples: chalk, clay

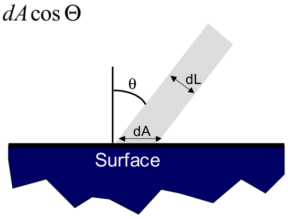


20

Diffuse Reflection

- How much light is reflected?
  - Depends on angle of incident light

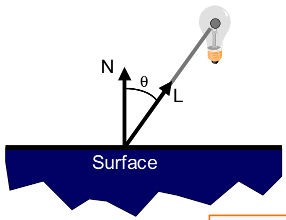
$$dL = dA \cos \Theta$$



21

Diffuse Reflection

- Lambertian model
  - cosine law (dot product)

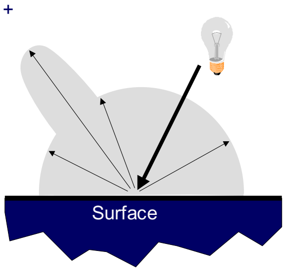


$$I_D = K_D (N \bullet L) I_L$$

22

OpenGL Reflectance Model

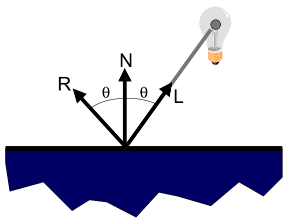
- Simple analytic model:
  - diffuse reflection +
  - specular reflection +
  - emission +
  - “ambient”



23

Specular Reflection

- Reflection is strongest near mirror angle
  - Examples: mirrors, metals

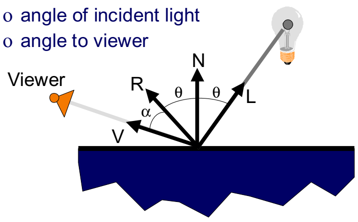


24

Specular Reflection

How much light is seen?

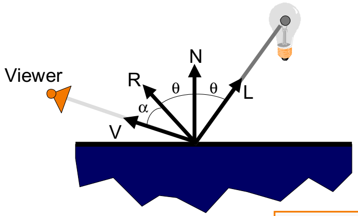
Depends on:
  - angle of incident light
  - angle to viewer



25

Specular Reflection

- Phong Model
  - o $\cos(\alpha)^n$

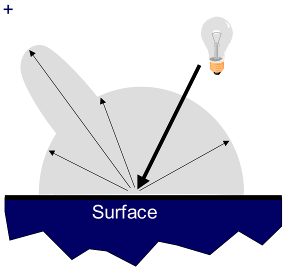


$$I_S = K_S (V \bullet R)^n I_L$$

26

OpenGL Reflectance Model

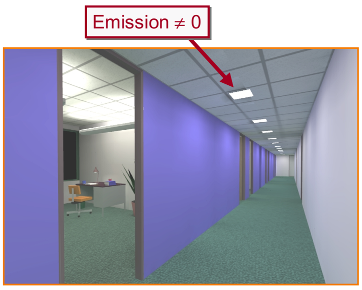
- Simple analytic model:
  - o diffuse reflection +
  - o specular reflection +
  - o emission +
  - o "ambient"



27

Emission

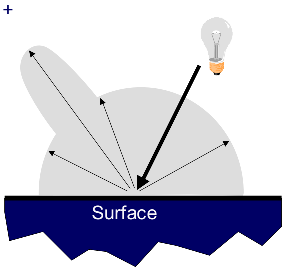
- Represents light eminating directly from polygon



28

OpenGL Reflectance Model

- Simple analytic model:
  - o diffuse reflection +
  - o specular reflection +
  - o emission +
  - o "ambient"



29

Ambient Term

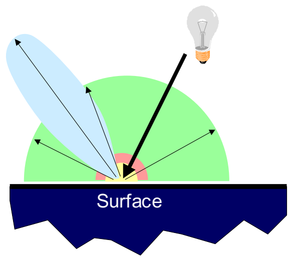
- Represents reflection of all indirect illumination

This is a total hack (avoids complexity of global illumination)!

30

OpenGL Reflectance Model

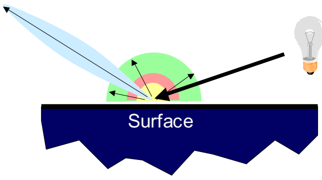
- Simple analytic model:
  - o diffuse reflection +
  - o specular reflection +
  - o emission +
  - o "ambient"



31

OpenGL Reflectance Model

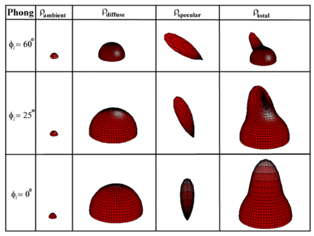
- Simple analytic model:
  - o diffuse reflection +
  - o specular reflection +
  - o emission +
  - o “ambient”



32

OpenGL Reflectance Model

- Sum diffuse, specular, emission, and ambient

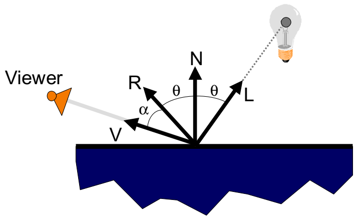


Leonard McMillan, MIT

33

Surface Illumination Calculation

- Single light source:

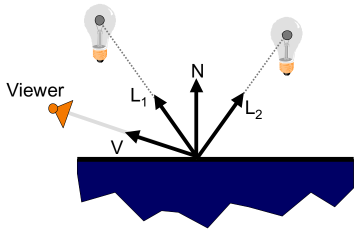


$$I = I_E + K_A I_{AL} + K_D (N \bullet L) I_L + K_S (V \bullet R)^n I_L$$

34

Surface Illumination Calculation

- Multiple light sources:



$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

35

Overview

- Direct Illumination
  - o Emission at light sources
  - o Scattering at surfaces
- Global illumination
  - o Shadows
  - o Transmissions
  - o Inter-object reflections

Global Illumination

36

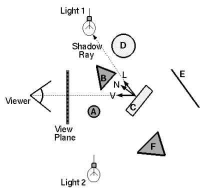
Global Illumination

Greg Larson

37

Shadows

- Shadow term tells if light sources are blocked
  - o Cast ray towards each light source L_i
  - o $S_i = 0$ if ray is blocked, $S_i = 1$ otherwise



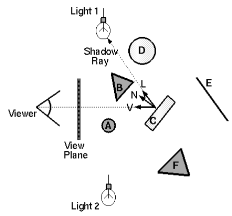
Shadow Term

$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n) S_L I_L$$

38

Ray Casting

- Trace primary rays from camera
  - o Direct illumination from unblocked lights only

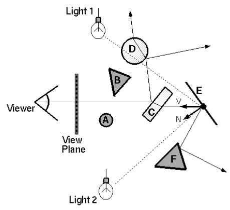


$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n) S_L I_L$$

39

Recursive Ray Tracing

- Also trace secondary rays from hit surfaces
  - o Global illumination from mirror reflection and transparency

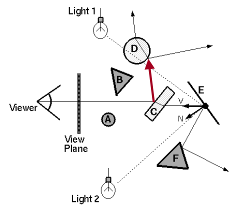


$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n) S_L I_L + K_S I_R + K_T I_T$$

40

Mirror reflections

- Trace secondary ray in mirror direction
  - o Evaluate radiance along secondary ray and include it into illumination model



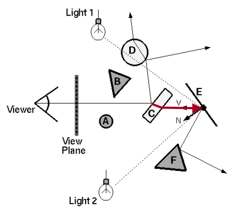
Radiance for mirror reflection ray

$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n) S_L I_L + K_S \boldsymbol{I_R} + K_T I_T$$

41

Transparency

- Trace secondary ray in direction of refraction
  - o Evaluate radiance along secondary ray and include it into illumination model



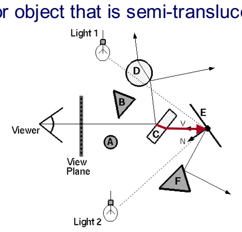
Radiance for refraction ray

$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n) S_L I_L + K_S I_R + K_T \boldsymbol{I_T}$$

42

Transparency

- Transparency coefficient is fraction transmitted
  - o $K_T = 1$ for translucent object, $K_T = 0$ for opaque
  - o $0 < K_T < 1$ for object that is semi-translucent



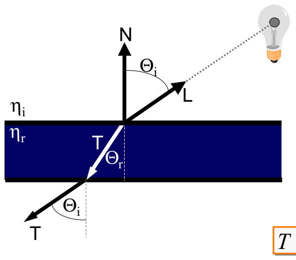
Transparency Coefficient

$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n) S_L I_L + K_S I_R + \boldsymbol{K_T} I_T$$

Refractive Transparency

43

- For thin surfaces, can ignore change in direction
  - o Assume light travels straight through surface



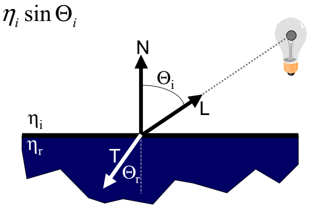
$T \cong -L$

Refractive Tranparency

44

- For solid objects, apply Snell's law:

$\eta_r \sin\Theta_r = \eta_i \sin\Theta_i$

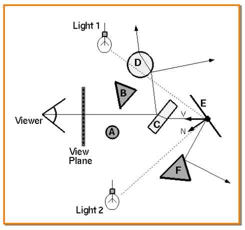


$$T = \left(\frac{\eta_i}{\eta_r}\cos\Theta_i - \cos\Theta_r\right)N - \frac{\eta_i}{\eta_r}L$$

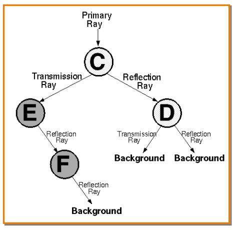
Recursive Ray Tracing

45

- Ray tree represents illumination computation



Ray traced through scene



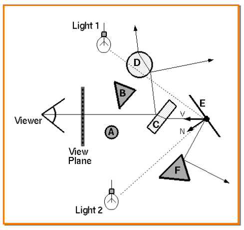
Ray tree

$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n)S_L I_L + K_S I_R + K_T I_T$$

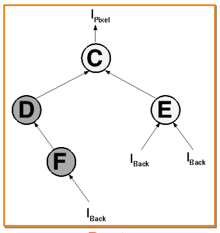
Recursive Ray Tracing

46

- Ray tree represents illumination computation



Ray traced through scene



Ray tree

$$I = I_E + K_A I_A + \sum_L (K_D(N \bullet L) + K_S(V \bullet R)^n)S_L I_L + K_S I_R + K_T I_T$$

Recursive Ray Tracing

47

- GetColor calls RayTrace recursively

```
Image RayTrace(Camera camera, Scene scene, int width, int height)
{
    Image image = new Image(width, height);
    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            Ray ray = ConstructRayThroughPixel(camera, i, j);
            Intersection hit = FindIntersection(ray, scene);
            image[i][j] = GetColor(scene, ray, hit);
        }
    }
    return image;
}
```

Summary

48

- Ray casting (direct Illumination)
  - o Usually use simple analytic approximations for light source emission and surface reflectance
- Recursive ray tracing (global illumination)
  - o Incorporate shadows, mirror reflections, and pure refractions

All of this is an approximation so that it is practical to compute

More on global illumination later!

49

Illumination Terminology

- Radiant power [flux] (Φ)
  - Rate at which light energy is transmitted (in Watts).
- Radiant Intensity (I)
  - Power radiated onto a unit solid angle in direction (in Watts/sr)
    - e.g.: energy distribution of a light source (inverse square law)
- Radiance (L)
  - Radiant intensity per unit projected surface area (in Watts/m^2sr)
    - e.g.: light carried by a single ray (no inverse square law)
- Irradiance (E)
  - Incident flux density on a locally planar area (in Watts/m^2)
    - e.g.: light hitting a surface along a
- Radiosity (B)
  - Exitant flux density from a locally planar area (in Watts/ m^2)