

### Light Transport

- Goal: Model the interaction of light with matter in a way that appears realistic (and is fast)

2  
5: Lighting & Shading

### Light Transport

- Assumptions:
  - Geometrical optics:
    - No diffraction, no polarization, no interference
  - Light travels in a straight line in a vacuum
    - No atmospheric scattering or refraction
    - No gravity effects
  - Discrete-wavelength approximation of color
    - Quantized approximation of dispersion and fluorescence
  - Superposition
    - No non-linear reflecting materials

3  
5: Lighting & Shading

### Bidirectional Reflectance

- Radiance L [W/(sr m<sup>2</sup>)]
  - radiant flux / (unit solid angle \* unit area)

4  
5: Lighting & Shading

### Bidirectional Reflectance

- Single incoming direction and single outgoing direction

$$L_r(\vec{\omega}_i, \vec{\omega}_r) = f_r(\vec{\omega}_i, \vec{\omega}_r) L_i(\vec{\omega}_i) \cos \theta_i d\vec{\omega}_i$$

↖ bidirectional reflectance distribution function

5  
5: Lighting & Shading

### Bidirectional Reflectance

- Integral over all incoming directions

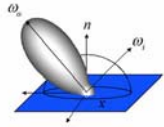
$$L_r(\vec{\omega}_r) = \int_{\Omega_i} f_r(\vec{\omega}_i, \vec{\omega}_r) L_i(\vec{\omega}_i) \cos \theta_i d\vec{\omega}_i$$

↖ incident hemisphere

6  
5: Lighting & Shading

**Bidirectional Reflectance**

- Bidirectional Reflectance Distribution Function (BRDF)
  - anisotropic (4 DOFs)  $f_r(\phi_i, \theta_i, \phi_r, \theta_r)$
  - isotropic (3DOFs)  $f_r(\phi_i - \phi_r, \theta_i, \theta_r)$



7  
5. Lighting & Shading


**Bidirectional Reflectance**

- Generalizations
  - anisotropic BRDF: 4 DOFs
  - spatially varying: +2
  - spectrally varying: +1
  - Fluorescence: +1
  - Phosphorescence: +1
  - Subsurface Scattering: +2

→ General BRDF is 11 dimensional function!!

8  
5. Lighting & Shading

**Subsurface Scattering**



© Henrik Wann Jensen

9  
5. Lighting & Shading


**Global Illumination**



© Henrik Wann Jensen

10  
5. Lighting & Shading

**Complex Materials and Lighting**



© Henrik Wann Jensen

11  
5. Lighting & Shading

**Real-World vs. OpenGL**

- Real world
  - complex computations
  - see optics textbooks, photorealistic rendering
- OpenGL
  - simplified model
  - ambient, diffuse and specular light sources and reflections
  - easy to tune
  - fast to compute

12  
5. Lighting & Shading

### Lighting

- Light sources
- Light reflection

The diagram illustrates light sources and reflection. At the top, three blue curved shapes represent light sources, each emitting yellow arrows representing light rays. Below, two more blue curved shapes represent surfaces reflecting light, with blue arrows showing the reflection of the light rays.

5 Lighting & Shading

### Ambient Light

- Scattered by environment
- Coming from all directions
- Reflection independent of
  - Camera position
  - Light position (no light position)
  - Surface orientation
- Reflected intensity:  $I = I_a k_a$ 
  - light source
  - material parameter

A blue ring is shown, representing ambient light. The text explains that ambient light is scattered by the environment and comes from all directions. The reflection is independent of camera position, light position, and surface orientation. The reflected intensity is given by the equation  $I = I_a k_a$ , where  $I_a$  is the light source and  $k_a$  is the material parameter.

5 Lighting & Shading

### Diffuse Reflection

- Directed light  $I_p$
- Reflection dependent on
  - orientation of surface
  - light source position
- Independent of
  - camera position (reflected equally in all directions)
- Reflected intensity:  $I = I_p k_d \cos \theta$   
 $I = I_p k_d (\mathbf{N} \cdot \mathbf{L})$

The diagram shows a blue ring representing diffuse reflection. Below it, a diagram illustrates a surface with a normal vector  $\mathbf{N}$  and a light vector  $\mathbf{L}$ . The angle between  $\mathbf{N}$  and  $\mathbf{L}$  is  $\theta$ . The reflected intensity is given by the equation  $I = I_p k_d \cos \theta$  or  $I = I_p k_d (\mathbf{N} \cdot \mathbf{L})$ .

5 Lighting & Shading

### Two Radiant Surface Patches

- Illumination  $B$  [lux]

The diagram shows two radiant surface patches, Source and Receiver, separated by a distance  $r$ . The Source patch has an area element  $dA_1$  and the Receiver patch has an area element  $dA_2$ . The illumination  $B$  is given by the equation  $B = \frac{dF}{dA_2}$ .

5 Lighting & Shading

### Diffuse Reflection

- Diffuse reflection scales with angle

The diagram shows a surface patch  $dA$  on Surface 1 and its projection  $dA / \cos \theta$  on Surface 2. The angle between the normal vector  $\mathbf{N}$  and the light vector  $\mathbf{L}$  is  $\theta$ . The illumination  $B$  is given by the equation  $B = \frac{dF}{dA / \cos \theta} = \cos \theta \frac{dF}{dA}$ .

illumination

5 Lighting & Shading

### A Simple Model

- Sum up ambient light and diffuse reflection:

$$I = I_a k_a + I_p k_d (\mathbf{N} \cdot \mathbf{L})$$

The diagram shows a blue ring representing the simple model. The equation  $I = I_a k_a + I_p k_d (\mathbf{N} \cdot \mathbf{L})$  is shown, representing the sum of ambient light and diffuse reflection.

5 Lighting & Shading

## Attenuation

- Quadratic attenuation due to spatial radiation

$$f_{att} = \frac{1}{d_L^2}$$

- A model often used in Graphics (OpenGL)

$$f_{att} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1\right)$$

```

gLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, c1);
gLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, c2);
gLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, c3);

```

- Include attenuation

$$I = I_a k_a + f_{att} I_p k_d (\mathbf{N} \cdot \mathbf{L})$$

19 5. Lighting & Shading

## Wavelength Dependency

- Colored light and reflection as functions of wavelength  $\lambda$

$$I_\lambda = I_a k_a O_{d_\lambda} + f_{att} I_{p_\lambda} k_d O_{d_\lambda} (\mathbf{N} \cdot \mathbf{L})$$

- Restriction to RGB (OpenGL)

$$I_R = I_{a_R} k_{a_R} + f_{att} I_{p_R} k_{d_R} (\mathbf{N} \cdot \mathbf{L})$$

$$I_G = I_{a_G} k_{a_G} + f_{att} I_{p_G} k_{d_G} (\mathbf{N} \cdot \mathbf{L})$$


$$I_B = I_{a_B} k_{a_B} + f_{att} I_{p_B} k_{d_B} (\mathbf{N} \cdot \mathbf{L})$$

⚠ Restriction of spectral sampling to RGB can lead to substantial color artifacts

20 5. Lighting & Shading

## Depth Cueing (Fog)

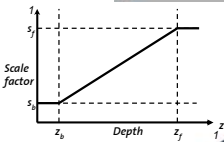
- Linear depth cue by blending with the color of the participating medium

$$I_z = s I_\lambda + (1-s) I_{atm}$$


- s is scaling factor

$$s = s_b + \frac{(z - z_b)(s_f - s_b)}{z_f - z_b}$$

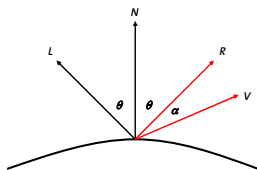
for  $z_b \leq z \leq z_f$



21 5. Lighting & Shading

## Specular Reflection

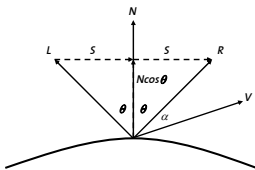
- Depends on angle between reflection and viewing ray



22 5. Lighting & Shading

## Reflected Ray

- Computed using simple vector algebra



$$\mathbf{R} = N \cos \theta + \mathbf{S}$$

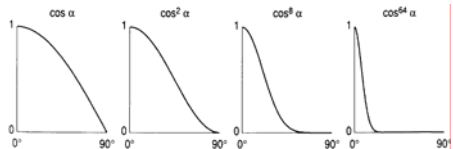
$$\mathbf{R} = 2N \cos \theta - \mathbf{L} = 2N(\mathbf{N} \cdot \mathbf{L}) - \mathbf{L}$$

$$\cos \alpha = \mathbf{R} \cdot \mathbf{V} = (2N(\mathbf{N} \cdot \mathbf{L}) - \mathbf{L}) \cdot \mathbf{V}$$

23 5. Lighting & Shading

## Phong Illumination Model

- Approximates specular reflection by cosine powers

$$I_\lambda = I_a k_a O_{d_\lambda} + f_{att} I_{p_\lambda} [k_d O_{d_\lambda} (\mathbf{N} \cdot \mathbf{L}) + k_s (\mathbf{R} \cdot \mathbf{V})^\alpha]$$


24 5. Lighting & Shading

### Ambient + Diffuse + Specular

Ambient      Diffuse      Specular

Ambient + Diffuse      Ambient + Diffuse + Specular

25  
5 Lighting & Shading

### Extensions

- Specular colors  

$$I_\lambda = I_{a_\lambda} k_a O_{d_\lambda} + f_{att} I_{p_\lambda} [k_d O_{d_\lambda} (\mathbf{N} \cdot \mathbf{L}) + k_s O_{s_\lambda} (\mathbf{R} \cdot \mathbf{V})^n]$$
- Halfway-Vector (faster)  

$$\cos^n \beta = (\mathbf{N} \cdot \mathbf{H})^n \quad \mathbf{H} = \frac{\mathbf{L} + \mathbf{V}}{|\mathbf{L} + \mathbf{V}|}$$
- Multiple Light Sources  

$$I_\lambda = I_{a_\lambda} k_a O_{d_\lambda} + \sum_{1 \leq i \leq m} f_{att_i} I_{p_{i\lambda}} [k_d O_{d_\lambda} (\mathbf{N} \cdot \mathbf{L}_i) + k_s O_{s_\lambda} (\mathbf{R}_i \cdot \mathbf{V})^n]$$

26  
5 Lighting & Shading

### Highlights

- $k_s/k_a$  determines highlighted surface regions

$k_s/k_a$

$\phi$

$k_a = 0.1$

27  
5 Lighting & Shading

### OpenGL

- Light source:
 

```
GLfloat light_pos[] = {x,y,z, 1};
GLfloat light_Ia[] = {IaR, IaG, IaB, IaA};
GLfloat light_Id[] = {IdR, IdG, IdB, IdA};
GLfloat light_Is[] = {IsR, IsG, IsB, IsA};
glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
glLightfv(GL_LIGHT0, GL_AMBIENT, light_Ia);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_Id);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_Is);
```
- Reflection (material):
 

```
GLfloat material_Ka[] = {kaR, kaG, kaB, kaA};
GLfloat material_Kd[] = {kdR, kdG, kdB, kdA};
GLfloat material_Ks[] = {ksR, ksG, ksB, ksA};
glMaterialfv(GL_FRONT, GL_AMBIENT, material_Ka);
glMaterialfv(GL_FRONT, GL_DIFFUSE, material_Kd);
glMaterialfv(GL_FRONT, GL_SPECULAR, material_Ks);
glMaterialfv(GL_FRONT, GL_SHININESS, Se);
```

28  
5 Lighting & Shading

### Light Position

- Position in world-coordinates
- Light stationary
 

```
gluLookAt(eyeX, eyeY, eyeZ,
          centerX, centerY, centerZ,
          upX, upY, upZ);
glLightfv(GL_LIGHT0, GL_POSITION, pos);
```
- Position in camera-coordinates
- Light moving with camera (headlight)
 

```
glLightfv(GL_LIGHT0, GL_POSITION, pos);
gluLookAt(eyeX, eyeY, eyeZ,
          centerX, centerY, centerZ,
          upX, upY, upZ);
```

29  
5 Lighting & Shading

### Light Sources

- Directed Spotlights
 
$$I_{L_\lambda} \cos^p \lambda$$

$$I_{L_\lambda} (-\mathbf{L} \cdot \mathbf{L}')^p$$

```
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, L');
glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, p);
```

30  
5 Lighting & Shading

## Intensity Diagrams

- Implemented in most modern graphics APIs

180° 135° 180° 135° 180° 135° 180° 135°  
 90° 45° 90° 45° 90° 45° 90° 45°  
 Uniformly radiating point source     $\cos \gamma$      $\cos^4 \gamma$      $\cos^{32} \gamma$

31  
5. Lighting & Shading

## Shading

- Shading requires many evaluations of a lighting model
- Questions:
  - where to evaluate?
  - when to evaluate?

32  
5. Lighting & Shading

## Shading Models

- Flat shading** (constant shading)
  - one color per primitive
- Gouraud shading**
  - linear interpolation of vertex intensities
- Phong shading**
  - linear interpolation of vertex normals
- Gouraud & Phong shading need vertex normals

33  
5. Lighting & Shading

## Gouraud Shading – Procedure

- Calculate face normals
- Calculate vertex normals
 
$$N_v = \frac{\sum_{i=1}^n N_i}{\left| \sum_{i=1}^n N_i \right|}$$
- Evaluate lighting model for each vertex
- Interpolate vertex intensities across primitive (bilinear interpolation)

34  
5. Lighting & Shading

## Gouraud Shading – Interpolation

- Linear interpolation on current scan line

$$I_a = I_1 - (I_1 - I_2) \frac{(y_1 - y_2)}{(y_1 - y_2)}$$

$$I_b = I_1 - (I_1 - I_3) \frac{(y_1 - y_2)}{(y_1 - y_3)}$$

$$I_p = I_b - (I_b - I_a) \frac{(x_p - x_a)}{(x_b - x_a)}$$

- Done during scan conversion
- Hardware acceleration possible

35  
5. Lighting & Shading

## Gouraud Shading – Quality

- Shading quality depends on size of projected primitives (relative to pixel size)

36  
5. Lighting & Shading

### Phong Shading

- Linear interpolation of normals (not intensities) on current scan line

- Evaluation of lighting model at each pixel
- More accurate but slower than Gouraud

37  
5. Lighting & Shading

### Shading Methods – Overview

- Shading in image space:
  - Flat shading (Constant shading)
  - Gouraud shading
- Shading in object / screen space:
  - Phong shading
- Impact on graphics hardware...

38  
5. Lighting & Shading

### Transparency

- Transparency done with  $\alpha$ -blending...
- Linearizing exponential attenuation of intensity in media

39  
5. Lighting & Shading

### Transparency – Basic Laws

- P1 filters intensity according to  $I'_{\lambda_2} = I_{\lambda_2} e^{-\alpha_1 \Delta t}$
- Linearization yields  $I'_{\lambda_2} = I_{\lambda_2} (1 - \alpha_1 \Delta t)$
- Emission of P1  $I'_{\lambda_1} = I_{\lambda_1} \alpha_1 \Delta t$
- Summing up (for unit length)
 
$$I_{\lambda} = I'_{\lambda_1} + I'_{\lambda_2} = I_{\lambda_1} \alpha_1 + I_{\lambda_2} (1 - \alpha_1)$$
- For N polygons ( **$\alpha$ -blending**)
 
$$I_{\lambda} = \sum_{i=1}^N \alpha_i I_{\lambda_i} \cdot \prod_{b=1}^{i-1} (1 - \alpha_b)$$

40  
5. Lighting & Shading