CS148: Introduction to Computer Graphics and Imaging

Rendering

Modeling & Simulating Appearance

Models
- Cameras
- Light sources
- Materials
  - Reflection
  - Texture

Lighting simulation
- Solving the rendering equation
- Ray tracing
Reflection

Materials

Plastic    Metal    Matte

From Apodaca and Gritz, *Advanced RenderMan*
Spheres [Matusik et al.]

Types of Reflection

Mirror
- Ideal reflection
- Reflection Law

Diffuse
- Matte
- Lambert’s Law

Specular
- Highlights and gloss
- Microfacet model
Lambert’s Law

![Diagram showing Lambert’s Law]

\[ E = E_L \cos \theta_L = E_L \hat{N} \cdot \hat{L} \]

Mirror: Ideal Specular Surface

Law of Reflection

![Diagram showing Law of Reflection]

\[ \hat{R} + (-\hat{I}) = 2 \cos \theta_i \hat{N} = -2(\hat{I} \cdot \hat{N})\hat{N} \]

\[ \hat{R} = \hat{I} - 2(\hat{I} \cdot \hat{N})\hat{N} \]
Microfacets: Glossy Reflection

\[
\cos \theta_H = \hat{N} \cdot \hat{H}
\]

\[
\hat{H} = \frac{\hat{L} + \hat{E}}{\sqrt{\hat{L} \cdot \hat{L} + \hat{E} \cdot \hat{E}}}
\]

Microfacet distribution

\[
(\cos \theta_H)^r = (\hat{N} \cdot \hat{H})^r
\]

Roughness \(r\)

GLSL

```glsl
vec3 N, L, V, H; float NdotL, NdotH;

vec4 Cd = gl_FrontMaterial.diffuse * gl_LightSource[0].diffuse;
vec4 Ca = gl_FrontMaterial.ambient * gl_LightSource[0].ambient;
Vec4 Cs = vec4(0.0);

N = normalize(gl_NormalMatrix * gl_Normal);
L = normalize(vec3(gl_LightSource[0].position));
NdotL = max(dot(N, L), 0.0);
if (NdotL > 0.0) {
    Cs = gl_FrontMaterial.specular * gl_LightSource[0].specular;
    H = normalize(gl_LightSource[0].halfVector.xyz);
    NdotH = max(dot(N, H), 0.0);
    Cs *= pow(NdotH, gl_FrontMaterial.shininess);
}

gl_FrontColor = Ca + NdotL * Cd + Cs;
```
Materials

Surface vs. Subsurface
Subsurface Scattering

Skin: Subsurface

Modeled by Stephen Stahlberg
Skin: Surface Only

Modeled by Stephen Stahlberg

Fiber Model

CS148 Lecture 19  Pat Hanrahan, Winter 2009
Sasquatch: Hair Modeling System

www.worley.com
CS148 Lecture 19

Stuart Aitken
Pat Hanrahan, Winter 2009
Lighting

Lighting Simulation

The Rendering Equation

Given a scene consisting of geometric primitives with material properties and a set of light sources, compute the illumination at each point on each surface.

Challenges

- Primitives complex: lights, materials, shapes
- Infinite number of light paths

Solution

- Ray tracing
Lighting Example: Cornell Box

Lighting: Diffuse Reflection

Surface Color

Diffuse Shading
Point Light Source
Lighting: Shadows

No Shadows
Point Light Source

Shadows
Point Light Source

Lighting: Soft Shadows

Hard Shadows
Point Light Source

Soft Shadows
Area Light Source
Lighting: Indirect Illumination

Program of Computer Graphics
Cornell University

Early Radiosity
Complex Indirect Illumination

Mies Courtyard House with Curved Elements

Modeling: Stephen Duck; Rendering: Henrik Wann Jensen

"Turing Test"

Measured

Simulated

Program of Computer Graphics
Cornell University
Three major reflection models
- Mirror reflection
  - Angle of incidence equals angle of reflection
- Diffuse reflection
  - Reflection proportional to the energy on the surface
- Specular reflection
  - Microfacet models

Materials
- Combination of diffuse and specular
- Natural materials: skin and hair, ??

Lighting
- Direct vs. indirect
- Point vs. area
- Shadows requires visibility query