CSE 167:
Introduction to Computer Graphics
Lecture #7: Color and Shading

Jürgen P. Schulze, Ph.D.
University of California, San Diego
Fall Quarter 2011
Announcements

- Homework project #3 due this Friday, October 14th
  - To be presented starting 1:30pm in lab 260
- Late submissions for project #2 accepted until this Friday
- Ted problem “Resource Unavailable” solved
Lecture Overview

**Color**
- Color reproduction on computer monitors
- Perceptually uniform color spaces

**Shading**
- Introduction
- Local shading models
Summary

- CIE color spaces are defined by matching curves
  - At each wavelength, matching curves give weights of primaries needed to produce color perception of that wavelength
  - CIE RGB matching curves determined using trisimulus experiment
- Each distinct color perception has unique coordinates
  - CIE RGB values may be negative
  - CIE XYZ values are always positive
CIE XYZ Color Space

Visualization

- Interpret XYZ as 3D coordinates
- Plot corresponding color at each point
- Many XYZ values do not correspond to visible colors
Chromaticity Diagram

- Project from XYZ coordinates to 2D for more convenient visualization

\[ x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z} \]

- Drop z-coordinate
Chromaticity Diagram

- Factor out luminance (perceived brightness) and chromaticity (hue)
  - \( x, y \) represent chromaticity of a color

\[
x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad 0 \leq x, y \leq 1
\]

- \( Y \) is luminance
- CIE xyY color space
- Reconstruct XYZ values from xyY

\[
X = \frac{Y}{y}x \quad Z = \frac{Y}{y}(1 - x - y)
\]
Chromaticity Diagram

- Visualizes x,y plane (chromaticities)
- Pure spectral colors on boundary

Colors shown do not correspond to colors represented by (x,y) coordinates!
Chromaticity Diagram

- Visualizes x,y plane (chromaticities)
- Pure spectral colors on boundary
- Weighted sum of any two colors lies on line connecting colors

Colors shown do not correspond to colors represented by (x,y) coordinates!
Chromaticity Diagram

- Visualizes x,y plane (chromaticities)
- Pure spectral colors on boundary
- Weighted sum of any two colors lies on line connecting colors
- Weighted sum of any number of colors lies in convex hull of colors (gamut)

Colors shown do not correspond to colors represented by (x,y) coordinates!
Gamut

- Any device based on three primaries can only produce colors within the triangle spanned by the primaries.
- Points outside gamut correspond to negative weights of primaries.

Gamut of CIE RGB primaries

Gamut of typical CRT monitor
RGB Monitors

- Given red, green, blue (RBG) values, what color will your monitor produce?
  - I.e., what are the CIE XYZ or CIE RGB coordinates of the displayed color?
- How are OpenGL RGB values related to CIE XYZ, CIE RGB?
- Often you don’t know!
  - OpenGL RGB ≠ CIE XYZ, CIE RGB
RGB Monitors

**Ideally:**

- We know XYZ values for RGB primaries
  \[(X_r, Y_r, Z_r)(X_g, Y_g, Z_g)(X_b, Y_b, Z_b)\]
- Monitor is linear
- RGB signal corresponds to weighted sum of primaries:

\[
\begin{bmatrix}
X_s \\
Y_s \\
Z_s
\end{bmatrix} = r \begin{bmatrix}
X_r \\
Y_r \\
Z_r
\end{bmatrix} + g \begin{bmatrix}
X_g \\
Y_g \\
Z_g
\end{bmatrix} + b \begin{bmatrix}
X_b \\
Y_b \\
Z_b
\end{bmatrix}
\]

\[
\begin{bmatrix}
X_s \\
Y_s \\
Z_s
\end{bmatrix} = \begin{bmatrix}
X_r & X_g & X_b \\
Y_r & Y_g & Y_b \\
Z_r & Z_g & Z_b
\end{bmatrix} \begin{bmatrix}
r \\
g \\
b
\end{bmatrix}
\]
RGB Monitors

- Given desired XYZ values, find rgb values by inverting matrix

\[
\begin{bmatrix}
X_s \\
Y_s \\
Z_s
\end{bmatrix}
\begin{bmatrix}
X_r & X_g & X_b \\
Y_r & Y_g & Y_b \\
Z_r & Z_g & Z_b
\end{bmatrix}^{-1}
= \begin{bmatrix}
r \\
g \\
b
\end{bmatrix}
\]

- Similar to change of coordinate systems for 3D points
RGB Monitors

In reality

- XYZ values for monitor primaries are usually not directly specified
  - Monitor brightness is adjustable

- Monitors are not linear

For typical CRT monitors

\[ I = V_s^\gamma \]

\[ \gamma \approx 2.2 \]
sRGB

- Standard color space, with standard conversion to CIE XYZ
- Designed to match RGB values of typical monitor under typical viewing conditions
  - If no calibration information available, it is best to interpret RGB values as sRGB
- sRGB is supported by OpenGL 2.0 with the ARB_framebuffer_sRGB extension
- For more details and transformation from CIE XYZ to sRGB: http://en.wikipedia.org/wiki/SRGB_color_space
Conclusions

- Color reproduction on consumer monitors is less than perfect
  - The same RGB values on one monitor look different than on another
  - Given a color in CIE XYZ coordinates, consumer systems do not reliably produce that color

- Need color calibration
  - Consumers do not seem to care
  - Standard for digital publishing, printing, photography
Lecture Overview

**Color**
- Color reproduction on computer monitors
- *Perceptually uniform color spaces*

**Shading**
- Introduction
- Local shading models
Perceptually Uniform Color Spaces

**Definition:**

Euclidean distance between color coordinates corresponds to perceived difference.

- CIE RGB, XYZ are not perceptually uniform:
  - Euclidean distance between RGB, XYZ coordinates does not correspond to perceived difference
MacAdam Ellipses

- Experiment (1942) to identify regions in CIE xy color space that are perceived as the same color
- Found elliptical areas, MacAdam ellipses
- In perceptually uniform color space, each point on an ellipse should have the same distance to the center
- Ellipses become circles
CIE L*,a*,b* (CIELAB)

- Most common perceptually uniform color space
  - L* encodes lightness
  - a* encodes position between magenta and green
  - b* encodes position between yellow and blue
- Uses asterisk (*) to distinguish from Hunter's Lab color space
- Conversion between CIE XYZ and CIELAB is non-linear
Further Reading

- **Wikipedia pages**

- **More details:**
  - CIE Color Space:
Lecture Overview

Color
- Color reproduction on computer monitors
- Perceptually uniform color spaces

Shading
- Introduction
- Local shading models
Shading

- Compute interaction of light with surfaces
- Requires simulation of physics
- “Global illumination”
  - Multiple bounces of light
  - Computationally expensive, minutes per image
  - Used in movies, architectural design, etc.
Global Illumination

- Covered by CSE168

(All non-teapot images courtesy of Prof. Wann Jensen)
Interactive Applications

- No physics-based simulation
- Simplified models
- Reproduce perceptually most important effects
- Local illumination
  - Only one bounce of light between light source and viewer
Rendering Pipeline

- **Scene data**
- **Modeling and viewing transformation**
- **Shading**
- **Projection**
- **Scan conversion, visibility**
- **Image**

- Position object in 3D
- Determine colors of vertices
  - Per vertex shading
- Map triangles to 2D
- Draw triangles
  - Per pixel shading
Lecture Overview

Color
- Color reproduction on computer monitors
- Perceptually uniform color spaces

Shading
- Introduction
- Local shading models
Local Illumination

- What gives a material its color?
- How is light reflected by a
  - Mirror
  - White sheet of paper
  - Blue sheet of paper
  - Glossy metal
Local Illumination

- Model reflection of light at surfaces
  - Assumption: no subsurface scattering
- Bidirectional reflectance distribution function (BRDF)
  - Given light direction, viewing direction, how much light is reflected towards the viewer
  - For any pair of light/viewing directions!
Local Illumination

Simplified model

- Sum of 3 components
- Covers a large class of real surfaces

diffuse + specular + ambient =
Local Illumination

Simplified model

- Sum of 3 components
- Covers a large class of real surfaces
Diffuse Reflection

- Ideal diffuse material reflects light equally in all directions
- View-independent
- Matte, not shiny materials
  - Paper
  - Unfinished wood
  - Unpolished stone
Diffuse Reflection

- Beam of parallel rays shining on a surface
  - Area covered by beam varies with the angle between the beam and the normal
  - The larger the area, the less incident light per area
  - Incident light per unit area is proportional to the cosine of the angle between the normal and the light rays
- Object darkens as normal turns away from light
- Lambert’s cosine law (Johann Heinrich Lambert, 1760)
- Diffuse surfaces are also called Lambertian surfaces
Diffuse Reflection

- Given
  - Unit surface normal $\mathbf{n}$
  - Unit light direction $\mathbf{L}$
  - Material diffuse reflectance (material color) $k_d$
  - Light color (intensity) $c_l$

- Diffuse color $c_d$ is:
  \[ c_d = c_l k_d (\mathbf{n} \cdot \mathbf{L}) \]
  Proportional to cosine between normal and light
Diffuse Reflection

Notes

- Parameters $k_d, c_l$ are r,g,b vectors
- Need to compute r,g,b values of diffuse color $c_d$ separately
- Parameters in this model have no precise physical meaning
  - $c_l$: strength, color of light source
  - $k_d$: fraction of reflected light, material color
Diffuse Reflection

- Provides visual cues
  - Surface curvature
  - Depth variation

Lambertian (diffuse) sphere under different lighting directions
OpenGL

- **Lights** (*glLight*)
  - Values for light: $(0, 0, 0) \leq c_l \leq (1, 1, 1)$
  - Definition: $(0,0,0)$ is black, $(1,1,1)$ is white

- **OpenGL**
  - Values for diffuse reflection
  - Fraction of reflected light: $(0, 0, 0) \leq k_d \leq (1, 1, 1)$

- **Consult OpenGL Programming Guide (Red Book)**
  - See course web site
Local Illumination

**Simplified model**

- Sum of 3 components
- Covers a large class of real surfaces
Specular Reflection

- **Shiny surfaces**
  - Polished metal
  - Glossy car finish
  - Plastics

- **Specular highlight**
  - Blurred reflection of the light source
  - Position of highlight depends on viewing direction
Specular Reflection

- Ideal specular reflection is mirror reflection
  - Perfectly smooth surface
  - Incoming light ray is bounced in single direction
  - Angle of incidence equals angle of reflection
Law of Reflection

- Angle of incidence equals angle of reflection

\[ \vec{R} + \vec{L} = 2 \cos \theta \vec{n} = 2(\vec{L} \cdot \vec{n})\vec{n} \]
\[ \vec{R} = 2(\vec{L} \cdot \vec{n})\vec{n} - \vec{L} \]
Specular Reflection

- Many materials are not perfect mirrors
  - Glossy materials

Glossy teapot
Glossy Materials

- Assume surface composed of small mirrors with random orientation (micro-facets)
- Smooth surfaces
  - Micro-facet normals close to surface normal
  - Sharp highlights
- Rough surfaces
  - Micro-facet normals vary strongly
  - Blurry highlight

<table>
<thead>
<tr>
<th>Polished</th>
<th>Smooth</th>
<th>Rough</th>
<th>Very rough</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Polished]</td>
<td>![Smooth]</td>
<td>![Rough]</td>
<td>![Very rough]</td>
</tr>
</tbody>
</table>
Glossy Surfaces

- Expect most light to be reflected in mirror direction
- Because of micro-facets, some light is reflected slightly off ideal reflection direction

Reflection
- Brightest when view vector is aligned with reflection
- Decreases as angle between view vector and reflection direction increases
Phong Model (Bui Tuong Phong, 1973)

- Specular reflectance coefficient $k_s$
- Phong exponent $p$
  - Greater $p$ means smaller (sharper) highlight

\[
c = k_s c_l (\mathbf{R} \cdot \mathbf{e})^p
\]
Phong Model

![Phong Model](image)
Blinn Model (Jim Blinn, 1977)

- Define unit halfway vector
  \[ \mathbf{h} = \frac{\mathbf{L} + \mathbf{e}}{||\mathbf{L} + \mathbf{e}||} \]

- Halfway vector represents normal of micro-facet that would lead to mirror reflection to the eye
Blinn Model

- The larger the angle between micro-facet orientation and normal, the less likely
- Use cosine of angle between them
- Shininess parameter
- Very similar to Phong

\[ c = k_s c_I (\mathbf{h} \cdot \mathbf{n})^s \]

49
Local Illumination

**Simplified model**
- Sum of 3 components
- Covers a large class of real surfaces

![Diagram of light components (diffuse, specular, ambient)]
Ambient Light

- In real world, light is bounced all around scene
- Could use global illumination techniques to simulate
- Simple approximation
  - Add constant ambient light at each point: $k_a c_a$
  - Ambient light color: $c_a$
  - Ambient reflection coefficient: $k_a$
- Areas with no direct illumination are not completely dark
Complete Blinn Model

- Blinn model with several light sources $I$
- All colors and reflection coefficients have separate values for red, green, blue

$$c = \sum_i c_{li} \left( k_d \left( L_i \cdot n \right) + k_s \left( H_i \cdot n \right)^{\beta} \right) + k_a c_a$$

diffuse + specular + ambient =
**BRDFs**

- Diffuse, Phong, Blinn models are instances of *bidirectional reflectance distribution functions* (BRDFs)
- For each pair of light directions $L$, viewing direction $e$, return fraction of reflected light
- Shading with general BRDF $f$

\[ c = \sum_i c_{li} f(L_i, e) \]

- Many forms of BRDFs in graphics, often named after inventors
  - Cook-Torrance
  - Ward
  - …
Next Lecture

- Light sources
- Shader programming:
  - Vertex shaders
  - Fragment shaders