Today

Color
- Review
- Color reproduction on standard monitors
- Perceptually uniform color spaces

Shading
- Introduction
- Local shading models

Light
- Electromagnetic waves [Maxwell 1862]
- Photons (tiny particles) [Planck 1900]

Simplified model in graphics
- Light rays carry a spectrum of energy

Color spaces
- Set of parameters describing a color sensation
  - Coordinate system for colors
- Could store full description of spectrum
  - Would be wasteful
- Human visual system
  - Three types of cones (light sensitive cells)
  - Expect three coordinates to be sufficient

CIE color spaces
- Based on trichromatic theory
  - Claims any color can be represented as a weighted sum of three primary colors
- Propose red, green, blue as primaries
- Goal
  - Given arbitrary color, determine the weights for the three primaries (tristimulus value) to reproduce that color sensation
  - Weights are “coordinates” of that color

Tristimulus experiment
- Determine tristimulus values for spectral colors experimentally
**Tristimulus experiment**

- Spectral primary colors were chosen
  - Blue (435.8nm), green (546.1nm), red (700nm)
- Matching curves for monochromatic target

- Negative values!

**Arbitrary spectrum**

- Arbitrary spectrum as sum of “monochromatic” spectra

  \[
  \begin{align*}
  L_0(\lambda) & \quad \text{“Monochromatic” spectra, width } h \\
  + & \quad L_1(\lambda) \\
  + & \quad L_2(\lambda) \\
  + & \quad L_3(\lambda) \\
  \sum_i L_i(\lambda) & \quad \text{Wavelength}
  \end{align*}
  \]

**Arbitrary spectrum**

- Arbitrary spectrum as sum of “monochromatic” spectra

  \[
  \begin{align*}
  L_0(\lambda) & \quad \text{“Monochromatic” spectra, width } h \\
  + & \quad L_1(\lambda) \\
  + & \quad L_2(\lambda) \\
  + & \quad L_3(\lambda) \\
  \sum_i L_i(\lambda) & \quad \text{Wavelength}
  \end{align*}
  \]

**CIE RGB values**

- Given spectrum, CIE RGB values are defined as
  \[
  \begin{align*}
  R &= \int \tilde{r}(\lambda)L(\lambda)d\lambda \\
  G &= \int \tilde{g}(\lambda)L(\lambda)d\lambda \\
  B &= \int \tilde{b}(\lambda)L(\lambda)d\lambda
  \end{align*}
  \]

- Matching curves for primaries \( \tilde{r}(\lambda), \tilde{g}(\lambda), \tilde{b}(\lambda) \)
- Problem: CIE RGB values can be negative

**CIE XYZ color space**

- Linear transformation of CIE RGB

  \[
  \begin{bmatrix}
  X \\
  Y \\
  Z
  \end{bmatrix} = \begin{bmatrix}
  1 & 1/1/3 & 1/3/3 \\
  1 & 1/1/3 & 1/3/3 \\
  1 & 1/1/3 & 1/3/3
  \end{bmatrix} \begin{bmatrix}
  R \\
  G \\
  B
  \end{bmatrix}
  \]

  \[
  \begin{bmatrix}
  G \\
  R \\
  P
  \end{bmatrix} = \begin{bmatrix}
  0.40 & 0.33 & 0.21 \\
  0.1767 & 0.8124 & 0.00 \text{?} \text{?} \text{?} \\
  0.00 & 0.01 & 0.99
  \end{bmatrix} \begin{bmatrix}
  R \\
  G \\
  B
  \end{bmatrix}
  \]

- Determined coefficients such that
  - \( Y \) corresponds to an experimentally determined brightness
  - No negative values in matching curves
  - White is XYZ=(1/3,1/3,1/3)

**CIE XYZ color space**

- No corresponding physical primaries

**Matching curves**

- Tristimulus values

  \[
  \begin{align*}
  X &= \int \tilde{x}(\lambda)L(\lambda)d\lambda \\
  Y &= \int \tilde{y}(\lambda)L(\lambda)d\lambda \\
  Z &= \int \tilde{z}(\lambda)L(\lambda)d\lambda
  \end{align*}
  \]

- Always positive!
Chromaticity diagram

- 2D visualization of CIE XYZ color space
  - Fix Y coordinate (brightness)
  - Project XYZ coordinates onto X+Y+Z=1 plane
  - Drop Z coordinate

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

Questions?

Today

Color
- Review
  - Color reproduction on standard monitors
  - Perceptually uniform color spaces

Shading
- Introduction
  - Local shading models

RGB monitors

- Given rgb 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

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

\[
\begin{bmatrix}
X_r \\
Y_r \\
Z_r
\end{bmatrix} = \begin{bmatrix}
X_r \\
Y_r \\
Z_r
\end{bmatrix} \begin{bmatrix}
r \\
g \\
b
\end{bmatrix}
\]
**RGB monitors**

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

\[
\begin{bmatrix}
X_a \\
Y_a \\
Z_a \\
\end{bmatrix}
\begin{bmatrix}
X_r & X_g & X_b \\
Y_r & Y_g & Y_b \\
Z_r & Z_g & Z_b \\
\end{bmatrix}^{-1}
\begin{bmatrix}
r \\
g \\
b \\
\end{bmatrix}
\]

- Similar to change of coordinate systems for 3D points

**In reality**

- XYZ values for monitor primaries are usually not directly specified
  - Monitor brightness is adjustable
- “White” depends on illumination due to environment
  - Monitors are not linear
  - Need gamma correction

**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’s best to interpret RGB values as sRGB
- sRGB is supported by OpenGL 2.1
- More details, exact transformation from CIE XYZ to sRGB

**Conclusions**

- Color reproduction on consumer monitors less than perfect
  - Same RGB values on one monitor look different than on another
  - Given 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

**Further reading**

- Wikipedia pages
- Other links

**Today**

**Color**

- Review
- Color reproduction on standard 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
- Conversion between CIE XYZ and CIELAB is non-linear

Questions?

Today
Color
- Review
- Color reproduction on standard 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
  - Movies, architectural design, etc.
Global illumination

- CSE168!

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

- Primitives
- Modeling and viewing transformation
- Shading
  - Position object in 3D
  - Determine colors of vertices
    - Per vertex shading
  - Map triangles to 2D
  - Draw triangles
    - Per pixel shading
- Projection
- Scan conversion, visibility
- Image

Today

Color

- Review
- Color reproduction on standard monitors
- Perceptually uniform color spaces

Shading

- Introduction
- Local shading models

Local illumination

- Model reflection of light at surfaces
- Appearance of surface can be computed if we know
  - Given light direction, viewing direction, how much light is reflected towards the viewer
  - For any pair of light/viewing directions!

Local illumination

- How is light reflected by a
  - Mirror
  - White sheet of paper
  - Blue sheet of paper
  - Glossy metal
Local illumination

Simplified model

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

\[
\text{diffuse} + \text{specular} + \text{ambient} = \text{total}\]

Diffuse reflection

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

\[
\text{Beam of parallel rays shining on a surface}
\]

- 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
- Diffuse surfaces are also called Lambertian surfaces

\[
\text{Cosine between normal and light}
\]

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

\[
\mathbf{c}_d = c_l k_d (\mathbf{n} \cdot \mathbf{L})
\]

Notes

- Parameters \( k_d, c_l \) are r,g,b vectors
- 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 \((0, 0, 0) \leq \alpha \leq (1, 1, 1)\)
    - \((0,0,0)\) 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 book
  - Online [http://fly.cc.fer.hr/%7Eunreal/theredbook/](http://fly.cc.fer.hr/%7Eunreal/theredbook/)

**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
  \[
  \begin{align*}
  \mathbf{R} & = \mathbf{L} - 2 \cos \theta \mathbf{n} - 2 (\mathbf{L} \cdot \mathbf{n}) \mathbf{n} \\
  \mathbf{R} & = 2 (\mathbf{L} \cdot \mathbf{n}) \mathbf{n} - \mathbf{L}
  \end{align*}
  \]
Specular reflection

- Many materials not quite perfect mirrors
  - Glossy materials

Glossy teapot

Glossy materials

- Assume surface composed of small mirrors with random orientation (microfacets)
- Smooth surfaces
  - Microfacet normals close to surface normal
  - Sharp highlights
- Rough surfaces
  - Microfacet normals vary strongly
  - Blurry highlights

Polished
Smooth
Rough
Very rough

Glossy surfaces

- Expect most light to be reflect in mirror direction
- Because of microfacets, 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

- Specular reflectance coefficient $k_s$
- Phong exponent $p$
  - Higher $p$, smaller (sharper) highlight

Phong model

\[ c = k_s c_I (R \cdot e)^p \]

Blinn model (Jim Blinn, 1977)

- Define unit halfway vector
  \[ h = \frac{L + e}{\|L + e\|} \]
- Halfway vector represents normal of microfacet that would lead to mirror reflection to the eye
Blinn model
- The larger the angle between microfacet orientation and normal, the less likely
- Use cosine of angle between them
- Shininess parameter $s$
- Very similar to Phong

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

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

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 $c_a$
  - Ambient reflection coefficient $k_a$
- Areas with no direct illumination are not completely dark

Complete model
- Blinn model with several light sources $i$

\[ c = \sum_i c_i (k_d (\mathbf{L}_i \cdot \mathbf{n}) + k_s (\mathbf{h}_i \cdot \mathbf{n})^s) + k_a c_a \]

Notes
- All colors, reflection coefficients have separate values for R,G,B
- Usually, ambient = diffuse coefficient
- For metals, specular = diffuse coefficient
  - Highlight is color of material
- For plastics, specular coefficient = $(x,x,x)$
  - Highlight is color of light

BRDFs
- Diffuse, Phong, Blinn models are instances of bidirectional reflectance distribution functions (BRDFs)
- For each pair of light directions $\mathbf{L}$, viewing direction $\mathbf{e}$, return fraction of reflected light
- Shading with general BRDF $f$
  \[ c = \sum_i c_i f (\mathbf{L}_i, \mathbf{e}) \]
- Many forms of BRDFs in graphics, often named after inventors
  - Cook-Torrance
  - Ward
  - ...
Next time

• More shading