University of British Columbia
CPSC 314 Computer Graphics
Jan-Apr 2008

Tamara Munzner

Color II, Lighting/Shading I

Week 7, Mon Feb 25

http://www.ugrad.cs.ubc.ca/~cs314/Vjan2008

News

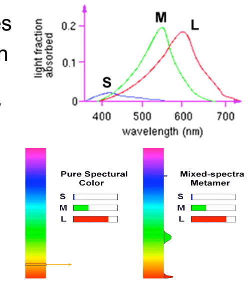
- I'm back!
  - including office hours Wed/Fri after lecture in lab
- this week
  - Fri 2/29: Homework 2 due 1pm sharp
  - Fri 2/29: Project 2 due 6pm
  - extra TA office hours in lab this week to answer questions
    - Tue 2-4 (usual lab 1-2)
    - Thu 2-4 (usual lab 10-11)
    - Fri 2-4 (usual lab 12-1)
- reminder: midterm next Fri Mar 7

News

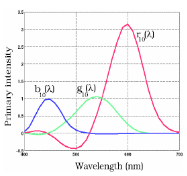
- Homework 1 returned today
  - average 84
- Project 1 face-to-face grading done
  - average 96
  - stragglers contact Cody, cjrobson@cs, ASAP
    - penalty for noshows, nosignups
- the glorious P1 Hall of Fame!

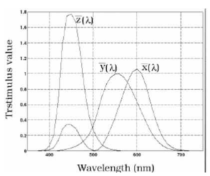
Review: Trichromacy and Metamers

- three types of cones
- color is combination of cone stimuli
  - metamer: identically perceived color caused by very different spectra



Review: Measured vs. CIE Color Spaces

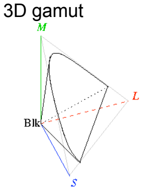


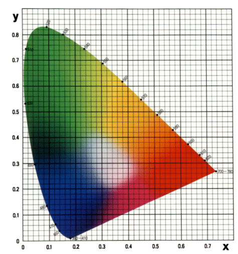


- measured basis
  - monochromatic lights
  - physical observations
  - negative lobes
- transformed basis
  - "imaginary" lights
  - all positive, unit area
  - Y is luminance, no hue
  - X, Z hue, no luminance

CIE Gamut and λ Chromaticity Diagram

- 3D gamut





- chromaticity diagram
  - hue only, no intensity

CIE "Horseshoe" Diagram Facts

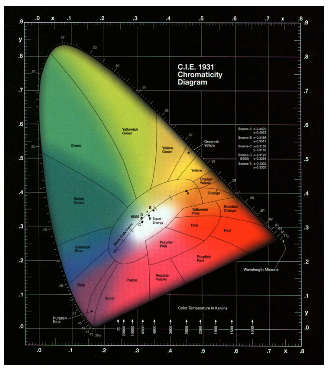
- all visible colors lie inside the horseshoe
  - result from color matching experiments
- spectral (monochromatic) colors lie around the border
  - the straight line between blue and red contains the purple tones
- colors combine linearly (i.e. along lines), since the xy-plane is a plane from a linear space

CIE "Horseshoe" Diagram Facts

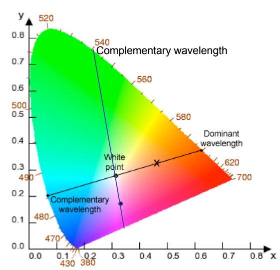
- a point C can be chosen as a white point corresponding to an illuminant
  - usually this point is of the curve swept out by the black body radiation spectra for different temperatures
  - relative to C, two colors are called complementary if they are located along a line segment through C, but on opposite sides (i.e C is an affine combination of the two colors)
  - the dominant wavelength of the color is found by extending the line from C through the color to the edge of the diagram
  - some colors (i.e. purples) do not have a dominant wavelength, but their complementary color does

CIE Diagram

- Blackbody curve
- Illumination:
  - Candle 2000K
  - Light bulb 3000K (A)
  - Sunset/ sunrise 3200K
  - Day light 6500K (D)
  - Overcast day 7000K
  - Lightning >20,000K

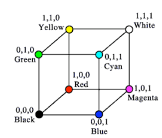


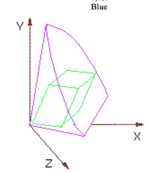
Color Interpolation, Dominant & Opponent Wavelength



RGB Color Space (Color Cube)

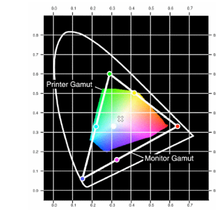
- define colors with (r, g, b) amounts of red, green, and blue
  - used by OpenGL
  - hardware-centric
  - describes the colors that can be generated with specific RGB light sources
- RGB color cube sits within CIE color space
  - subset of perceivable colors
  - scaled, rotated, sheared cube

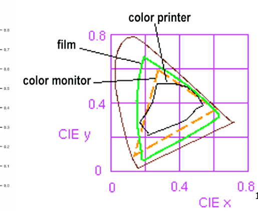




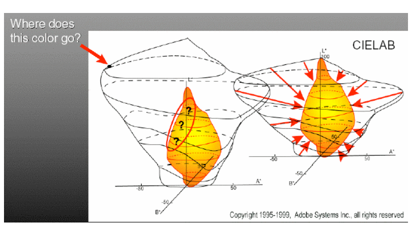
Device Color Gamuts

- use CIE chromaticity diagram to compare the gamuts of various devices
  - X, Y, and Z are hypothetical light sources, not used in practice as device primaries





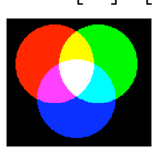
Gamut Mapping

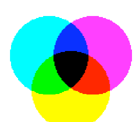


Additive vs. Subtractive Colors

- additive: light
  - monitors, LCDs
  - RGB model
- subtractive: pigment
  - printers
  - CMY(K) model

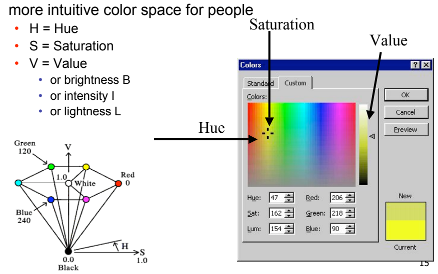
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$





HSV Color Space

- more intuitive color space for people
  - H = Hue
  - S = Saturation
  - V = Value
    - or brightness B
    - or intensity I
    - or lightness L



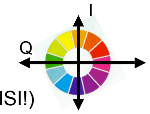
HSI/HSV and RGB

- HSV/HSI conversion from RGB
  - hue same in both
  - value is max, intensity is average

$$H = \cos^{-1}\left[\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right]$$ if (B > G), H = 360 - H

- HSI: $S = 1 - \frac{\min(R,G,B)}{I}$ $\quad I = \frac{R+G+B}{3}$
- HSV: $S = 1 - \frac{\min(R,G,B)}{V}$ $\quad V = \max(R,G,B)$

YIQ Color Space



- color model used for color TV
  - Y is luminance (same as CIE)
  - I & Q are color (not same I as HSI!)
  - using Y backwards compatible for B/W TVs
  - conversion from RGB is linear

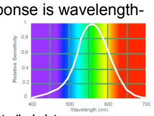
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- green is much lighter than red, and red lighter than blue

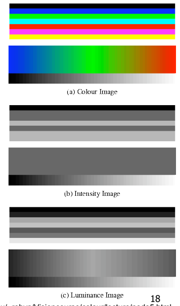
17

HSV Does Not Encode Luminance

- luminance
  - Y of YIQ
  - 0.299R + 0.587G + 0.114B
- luminance takes into effect that eye spectral response is wavelength-dependent



- value/intensity/brightness
  - I/V/B of HSI/HSV/HSB
  - 0.333R + 0.333G + 0.333B
  - lose information!



http://www.yorku.ca/eye/photopik.htm

www.csse.uwa.edu.au/~robyn/Visioncourse/colour/lecture/node5.html

18

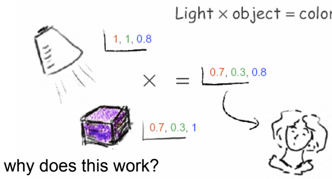
Luminance and Gamma Correction

- humans have nonlinear response to brightness
  - luminance 18% of X seems half as bright as X
- thus encode luminance nonlinearly: perceptually uniform domain uses bits efficiently
  - high quality with 8 bits, instead of 14 bits if linear
- monitors, sensors, eye all have different reponses
  - CRT monitors inverse nonlinear, LCD panels linear
  - characterize by gamma
    - displayedIntensity = a^{γ} (maxIntensity)
- gamma correction
  - displayedIntensity = $\left(a^{1/\gamma}\right)^{\gamma}$ (maxIntensity) = a (maxIntensity)
  - gamma for CRTs around 2.4

19

RGB Component Color (OpenGL)

- simple model of color using RGB triples
- component-wise multiplication
  - (a0,a1,a2) * (b0,b1,b2) = (a0*b0, a1*b1, a2*b2)



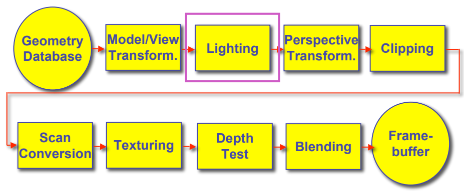
- why does this work?
  - because of light, human vision, color spaces, ...

20

Lighting I

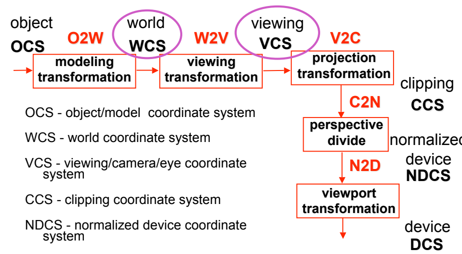
21

Rendering Pipeline



22

Projective Rendering Pipeline



OCS - object/model coordinate system

WCS - world coordinate system

VCS - viewing/camera/eye coordinate system

CCS - clipping coordinate system

NDCS - normalized device coordinate system

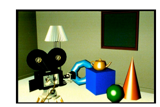
DCS - device/display/screen coordinate system

23

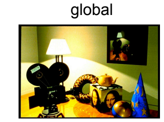
Goal

- simulate interaction of light and objects
- fast: fake it!
  - approximate the look, ignore real physics
- local model: interaction of each object with light
  - vs. global model: interaction of objects with each other

local



global



24

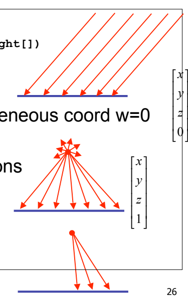
Illumination in the Pipeline

- local illumination
  - only models light arriving directly from light source
  - no interreflections or shadows
    - can be added through tricks, multiple rendering passes
- light sources
  - simple shapes
- materials
  - simple, non-physical reflection models

25

Light Sources

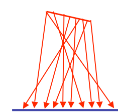
- types of light sources
  - `glLightfv(GL_LIGHT0,GL_POSITION,light[])`
- directional/parallel lights
  - real-life example: sun
  - infinitely far source: homogeneous coord w=0
- point lights
  - same intensity in all directions
- spot lights
  - limited set of directions:
    - point+direction+cutoff angle



26

Light Sources

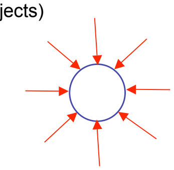
- area lights
- light sources with a finite area
- more realistic model of many light sources
- not available with projective rendering pipeline (i.e., not available with OpenGL)



27

Light Sources

- ambient lights
- no identifiable source or direction
- hack for replacing true global illumination
  - (diffuse interreflection: light bouncing off from other objects)



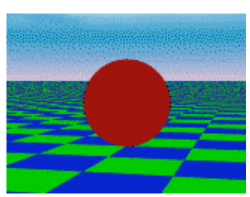
28

Diffuse Interreflection

29

Ambient Light Sources

- scene lit only with an ambient light source



Light Position Not Important

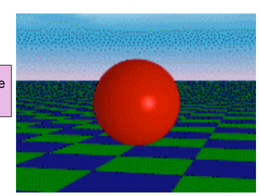
Viewer Position Not Important

Surface Angle Not Important

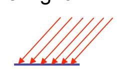
30

Directional Light Sources

- scene lit with ambient and directional light



Surface Angle Important



Light Position Not Important

Viewer Position Not Important

31

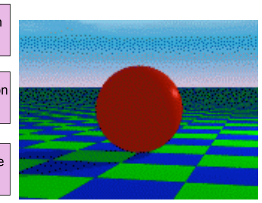
Point Light Sources

- scene lit with ambient and point light source

Light Position Important

Viewer Position Important

Surface Angle Important



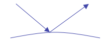
32

Light Sources

- geometry: positions and directions
  - coordinate system used depends on when you specify
  - standard: world coordinate system
    - effect: lights fixed wrt world geometry
    - demo: http://www.xmission.com/~nate/tutors.html
  - alternative: camera coordinate system
    - effect: lights attached to camera (car headlights)
  - points and directions undergo normal model/view transformation
- illumination calculations: camera coords

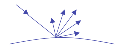
Types of Reflection

- *specular* (a.k.a. *mirror* or *regular*) reflection causes light to propagate without scattering.



- *diffuse* reflection sends light in all directions with equal energy.

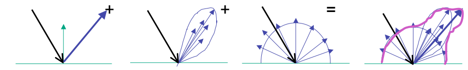
- *glossy/mixed* reflection is a weighted combination of specular and diffuse.



Specular Highlights

Reflectance Distribution Model

- most surfaces exhibit complex reflectances
  - vary with incident and reflected directions.
  - model with combination

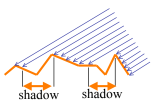


specular + glossy + diffuse = reflectance distribution

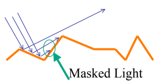
Surface Roughness

- at a microscopic scale, all real surfaces are rough

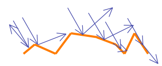
- cast shadows on themselves



- "mask" reflected light:



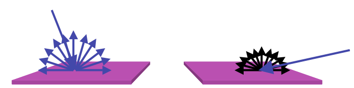
Surface Roughness



- notice another effect of roughness:
  - each "microfacet" is treated as a perfect mirror.
  - incident light reflected in different directions by different facets.
  - end result is mixed reflectance.
    - smoother surfaces are more specular or glossy.
    - random distribution of facet normals results in diffuse reflectance.

Physics of Diffuse Reflection

- ideal diffuse reflection
  - very rough surface at the microscopic level
    - real-world example: chalk
  - microscopic variations mean incoming ray of light equally likely to be reflected in any direction over the hemisphere
  - what does the reflected intensity depend on?

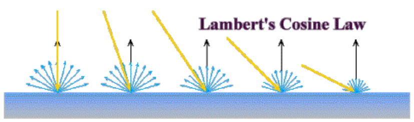


Lambert's Cosine Law

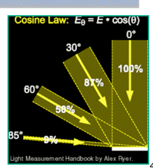
- ideal diffuse surface reflection

  the energy reflected by a small portion of a surface from a light source in a given direction is proportional to the cosine of the angle between that direction and the surface normal
- reflected intensity
  - independent of viewing direction
  - depends on surface orientation wrt light
- often called Lambertian surfaces

Lambert's Law

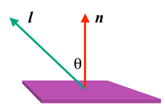


intuitively: cross-sectional area of the "beam" intersecting an element of surface area is smaller for greater angles with the normal.



Computing Diffuse Reflection

- depends on angle of incidence: angle between surface normal and incoming light
  - $I_{diffuse} = k_d \, I_{light} \cos \theta$
- in practice use vector arithmetic
  - $I_{diffuse} = k_d \, I_{light} \, (\mathbf{n} \bullet \mathbf{l})$



- **always normalize vectors used in lighting!!!**
  - **n**, **l** should be unit vectors
- scalar (B/W intensity) or 3-tuple or 4-tuple (color)
  - k_d: diffuse coefficient, surface color
  - I_{light}: incoming light intensity
  - $I_{diffuse}$: outgoing light intensity (for diffuse reflection)

Diffuse Lighting Examples

- Lambertian sphere from several lighting angles:

- need only consider angles from 0° to 90°
  - *why?*
  - *demo: Brown exploratory on reflection*
  - http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/reflection2D/reflection_2d_java_browser.html