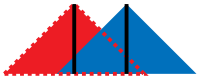
CS559 – Lecture 7
Color, Quantization

These are course notes (not used as slides)
Written by Mike Gleicher, Sept. 2005
With some slides adapted from the notes of Stephen Chenney

Preliminary version (before class)

Faking Colors

- Metamers allow for faking
- Two different overlapping cones respond
  - Some of each color?
  - Some of the in-between color
- Can fake responses using N "point" colors
- 2 cones = 2 frequencies
- Get either cone, or anything in overlap
- Colors outside of overlap can't be faked

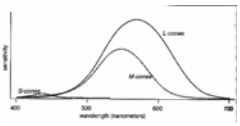


Gamut

- The range of colors that a device can represent
  - Perceptual range

(normal) Human Vision

- 3 types of Cones
  - S (short wavelength) cones
  - M (mid wavelength) cones
  - L (long wavelength) cones
- Sortof RGB, but not quite
- Lots of overlap
- Far fewer S cones than L and M



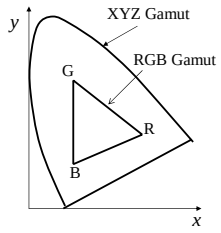
Different Sensitivities

- Convert to gray requires scaling for sensitivities
- R = 0.212671 * Y
- G = 0.715160 * Y
- B = 0.072169 * Y.

Perceptual Color Space

- Choose 3 primaries that do span human vision
  - Complete Gamut – can recreate any color
  - Not physically realizable (since has negative energies)
- CIE XYZ
  - Y is "lightness" – intensity w/o color
  - XZ are color directions

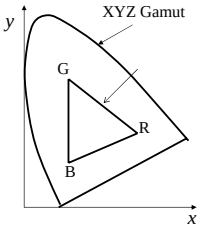
Determining Gamuts



- Gamut: The range of colors that can be represented or reproduced
- Plot the matching coordinates for each primary. eg R, G, B
- Region contained in triangle (3 primaries) is gamut
- Really, it's a 3D thing, with the color cube distorted and embedded in the XYZ gamut

Gamut Analysis

- Space of colors a device can reproduce depends on primaries
- Device reproduce linear combinations of primaries = space inside of points
- Different devices have different ranges
  - Print with more inks
  - Films with different formulations



Is RGB good enough?

- Sortof – gets close to all colors
  - Need better gamut
- No
  - Inconvenient for talking about color
  - Perceptually non-linear
  - Can't get really vivid colors
    - Purples are particularly bad
    - Can't be RGV – since violet sensitivity isn't good
    - Old film had different gamuts
      - Robin hood in technicolor

Other Color Systems: YCC

- Y = Luminence
  - Could be R+G+B
  - Better to be .3R + .6G + .1B
- Redundant – so send just 2 colors
  - Or send color differences: Y-R, Y-G
- Why?
  - Video: luminance is most important, subsample chroma
  - Perceptually more uniform since corrected for sensitivity
  - Start to separate color (direction in 2D)

Subtractive Color

- Printers combine inks that filter light
  - Remove colors
- So far additive
  - Black + red + green = yellow
- Ink is subtractive
  - White – red = cyan, White-green=magenta, white-blue=yellow
- Use "subtractive primaries"
  - Cyan, Magenta, Yellow

Artist – Centric Systems

- Hue = "name" of color
  - Red, orange, yellow, …
  - Color wheel
    - Complements add up to white
- Saturation = purity
- Value = luminence
- HSV (hexcone) vs. HLS (double hexcone)
  - RGB Color Cube viewed from the end
- Cone shape
  - Value is zero, hard to talk about color
- More convenient way to talk about color (for artists)

Where color gets messy…

- Color reproduction is hard
- When you see something on a monitor, does it look like the real thing? (shopping)
  - When you buy a real object?
  - When you print it?

Representing Color

- RGB
  - Store brightness for each channel
  - 8 bits argument (1% difference, 100:1 ~~ 400)
- Color Tables
  - A small table of integers->color
  - Store small integers for each pixel
  - Used a lot in old days (24 bits of frame buffer was a lot of memory!)
  - Still useful in some settings
    - Animate color tables, restrict pallete, …
  - Lots of algorithms for picking sets of colors
    - Median Cut is the most famous

Quantization

- What happens when we want smaller numbers of values?
  - Black and white for printing
  - Limited color pallete
- Old problem
  - Printing
  - Artists (pen and ink drawing)

Thresholding

- Threshold – pick value / above or below
- Each pixel picks nearest value
  - 49% looks the same as 1%
  - 49% looks very different than 51%
- Better: trade spatial resolution for value resolution
  - Brain blurs stuff together anyway
  - Art example: hatching to show “gray”

Dithering

- Add some random noise
- 50% + noise -> half black, half white
- Values at extreme less likely to get changed
- Eye doesn’t mind noise as much as it does blocky edges