CS 519: Scientific Visualization

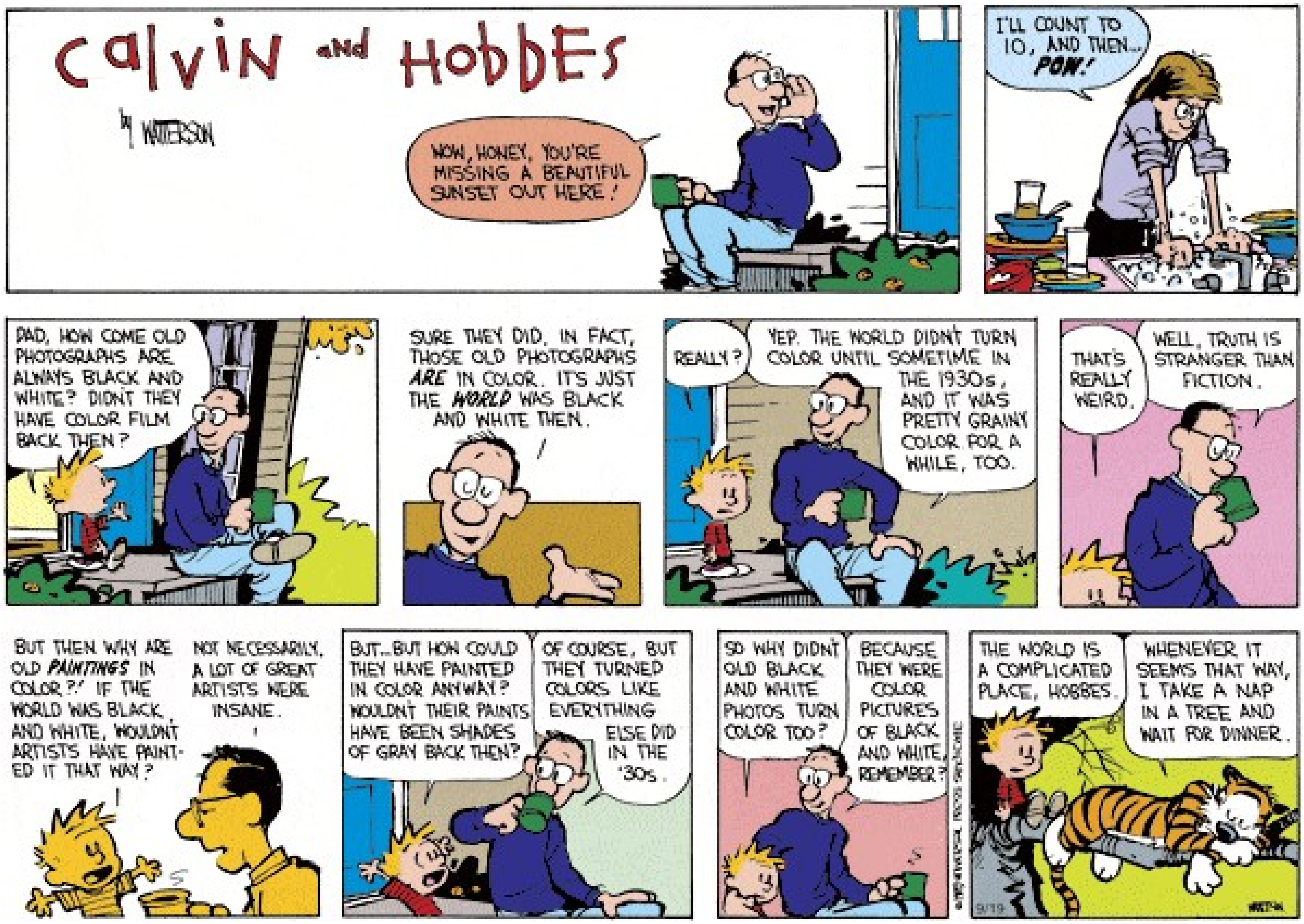
Color

Eric Shaffer

Some slides adapted Alexandru Telea, *Data Visualization Principles and Practice*

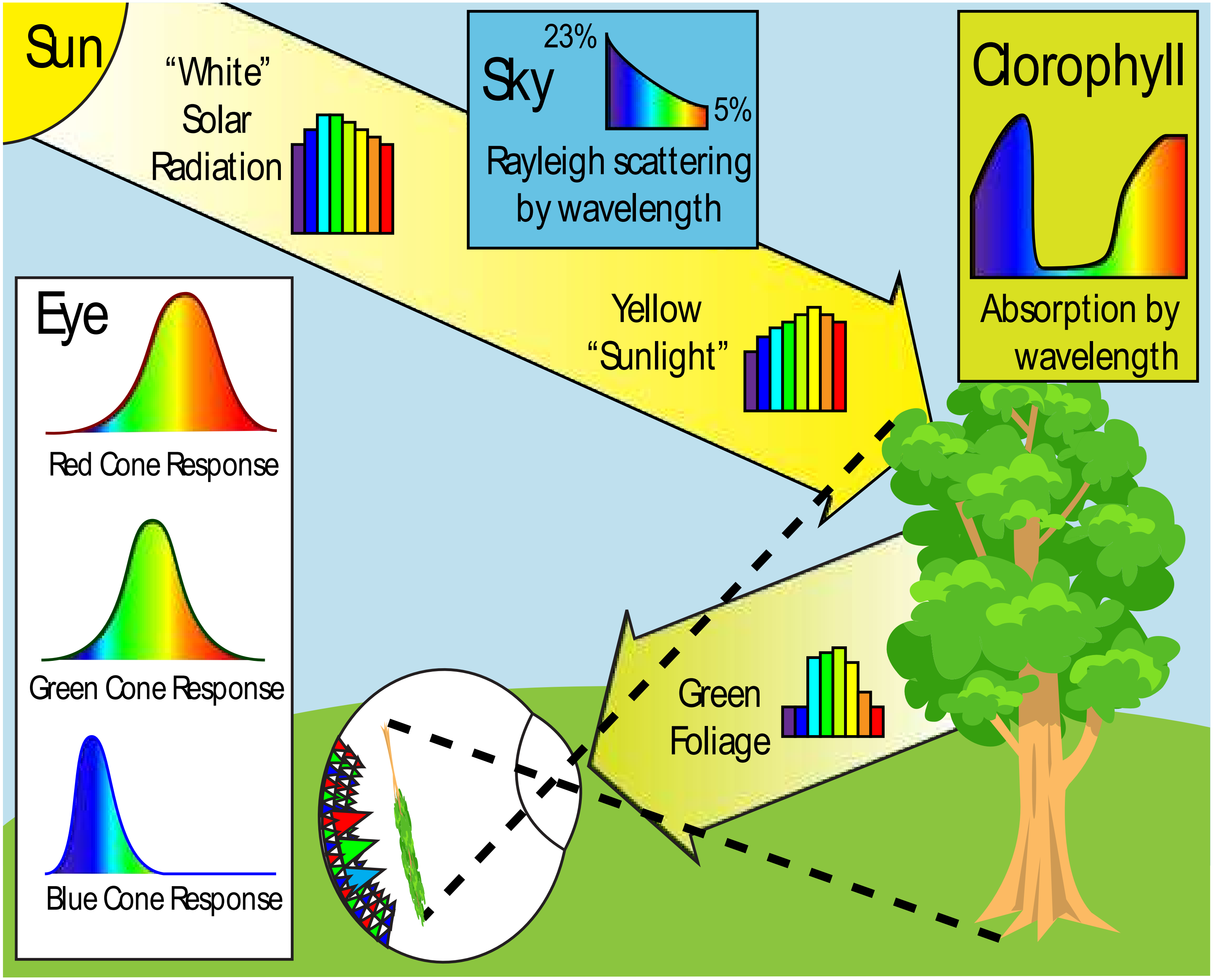
Some slides adapted from *Information Visualization* by Colin Ware

Rainbow versus Black and White



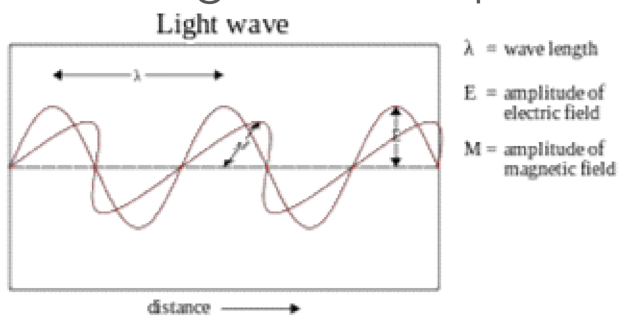
Color

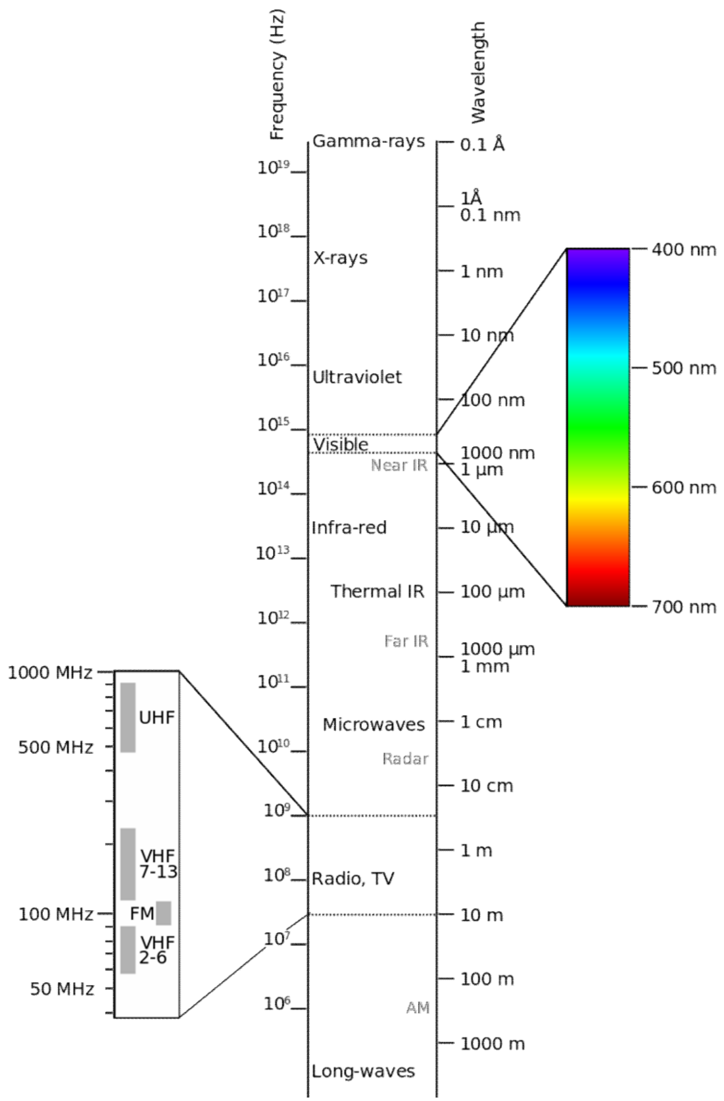
- Color is a perceptual phenomenon
- A frequency spectrum of light is a physical phenomenon
- In computer graphics, we need to specify colors
  - We define "color spaces" in which points correspond to colors
  - We can then work with colors mathematically
- Ideally, a color space would allow us to specify any color that humans can perceive…


Sun
"White"
Solar
Radiation
Sky
23%
5%
Rayleigh scattering
by wavelength
Clorophyll
Absorption by
wavelength
Yellow
"Sunlight"
Eye
Red Cone Response
Green Cone Response
Blue Cone Response
Green
Foliage

Light and Color

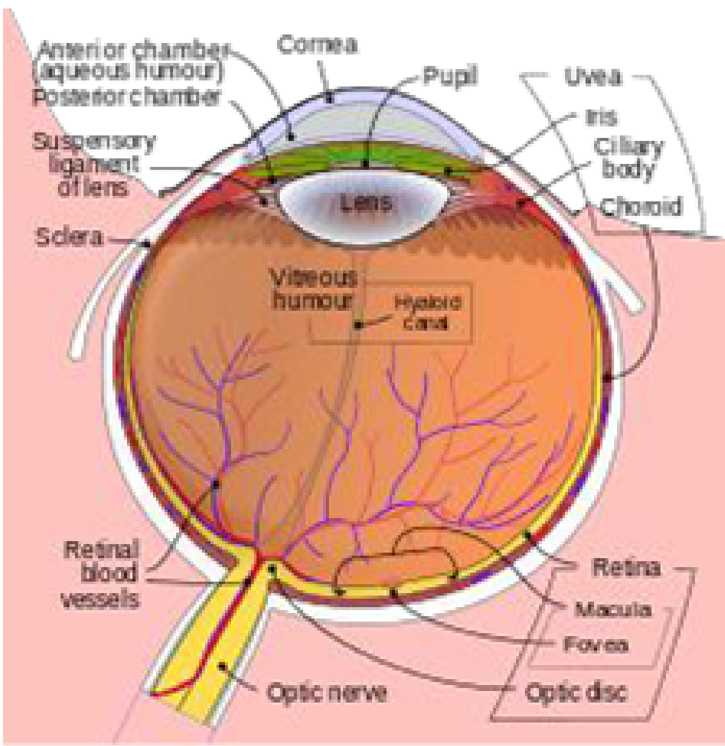
- Color is a perceptual phenomenon
  - Response of the human visual system to light…and other factors
- Light is a physical phenomenon
  - Electromagnetic radiation visible to the human eye
  - Emitted in quanta called photons
  - Has wavelength and amplitude

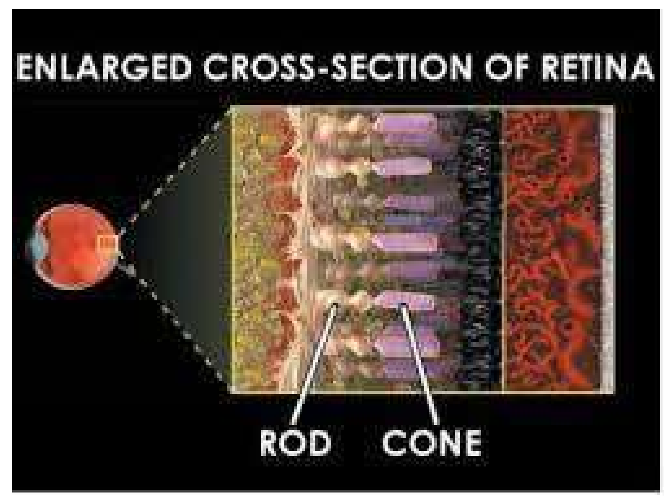




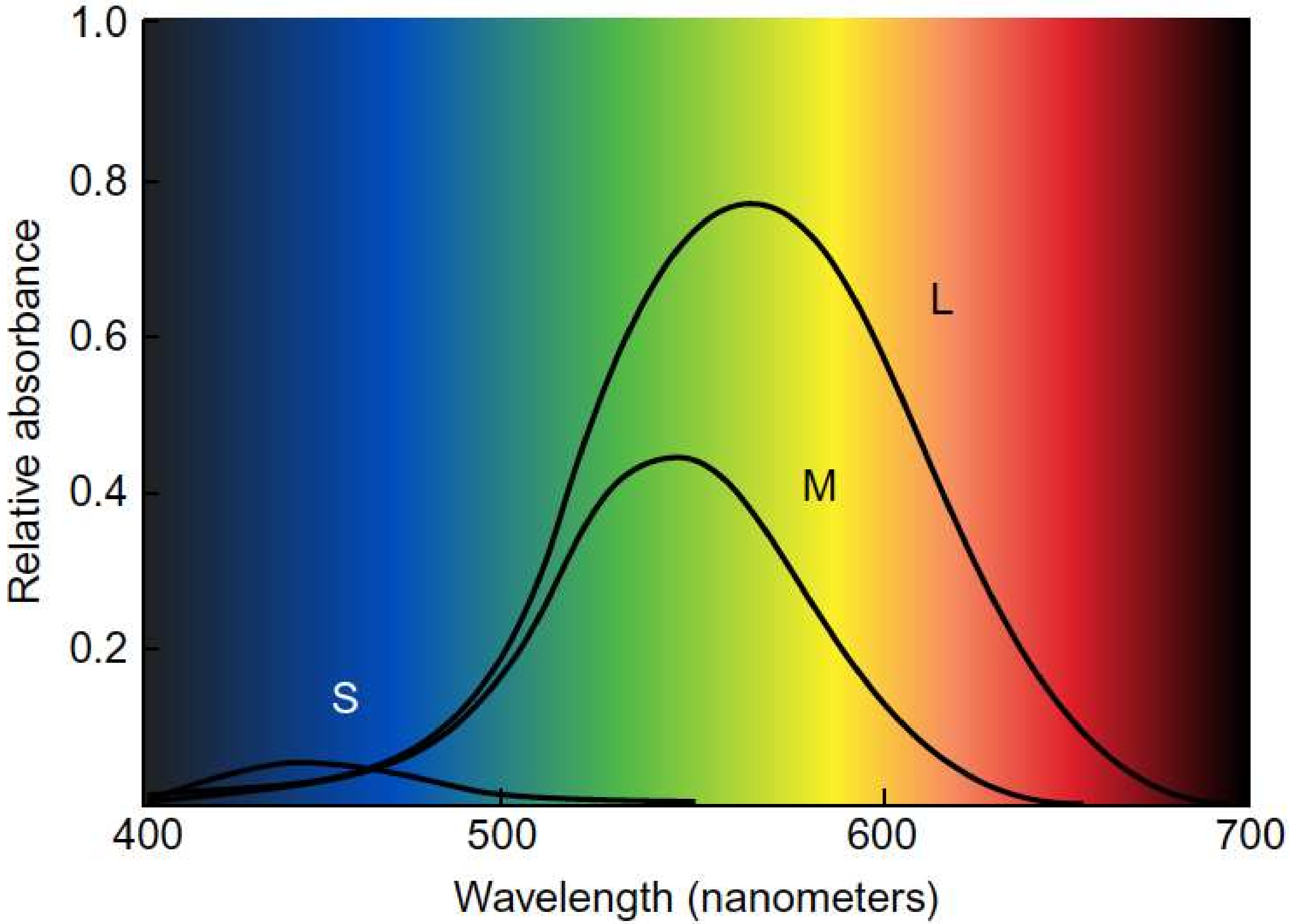
Photometry

- Perceptual study of light
- Color depends on interaction of light and the human eye



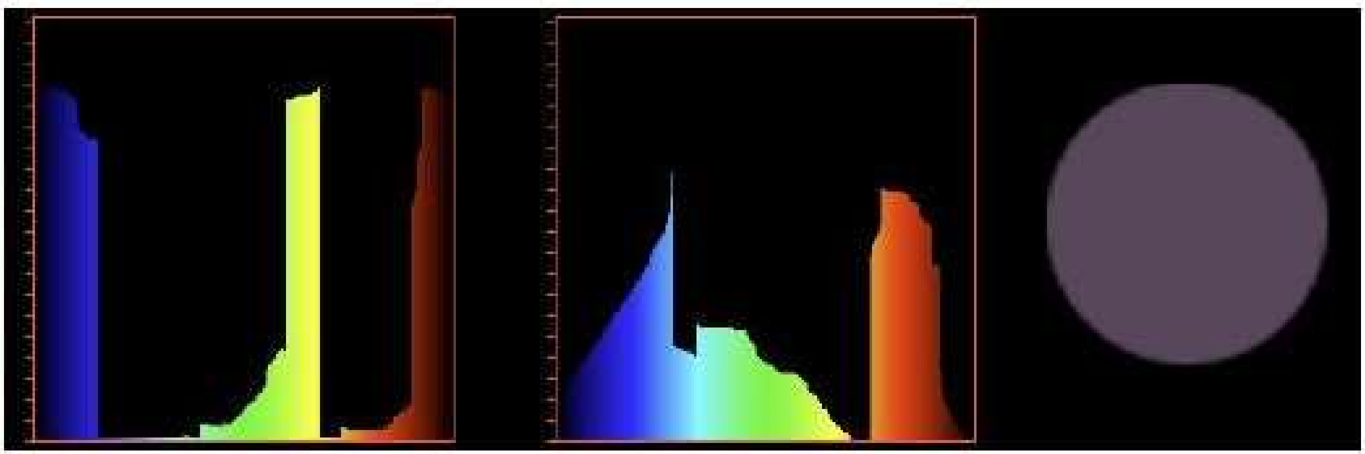


Cone Response



Tristimulus Theory

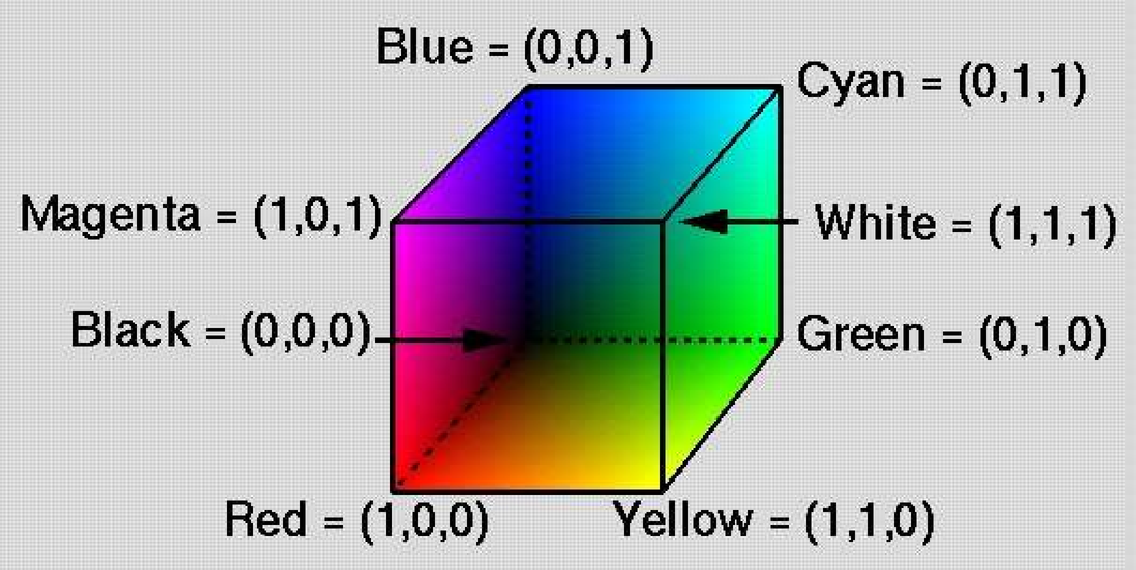
- 3 cone types suggest 3 parameters describe all colors
- Two different spectral distributions can appear the same
  - metamers

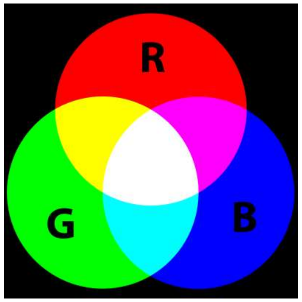


Different spectra can appear the same color (Hughes, Bell and Doppelt)

RGB Additive Color

- Red, Green, Blue
- Color model used in luminous displays (CRT, plasma, LCD)
- Physically linear
- Perceptually logarithmic
- Additive
- Designed to stimulate each kind of cone





CMY Color Space: Subtractive Color

- Cyan, Magenta, Yellow
- Color model used in pigments and reflective materials (ink,paint)
- Grade school color rules

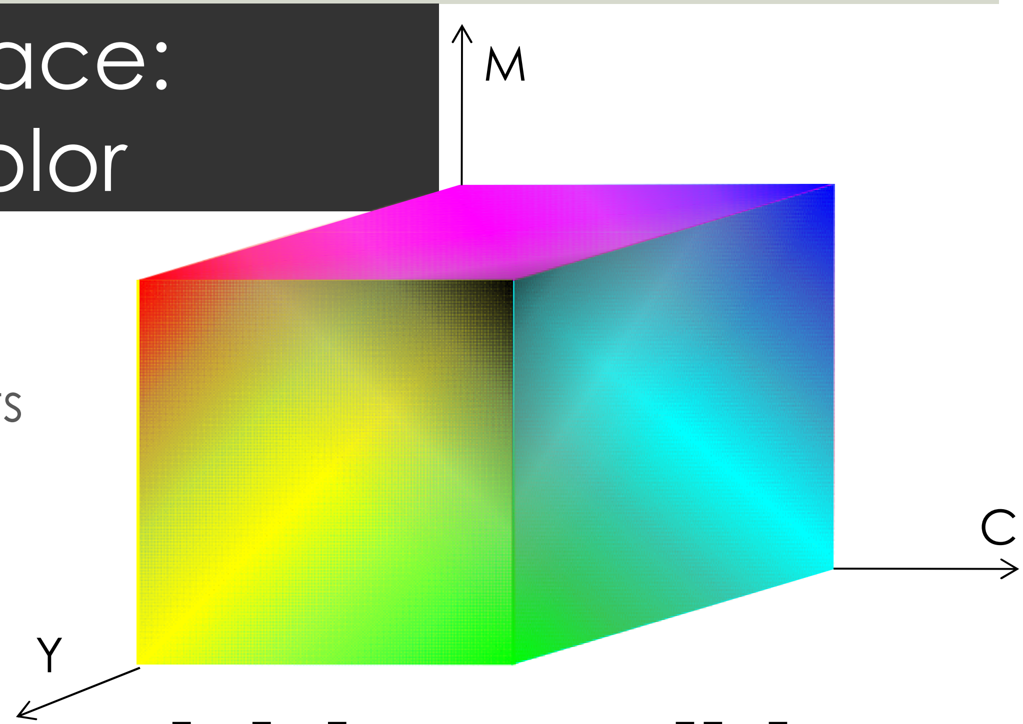
  Blue + Yellow = Green?

  Cyan + Yellow = Green
- Also CMYK (blacK)

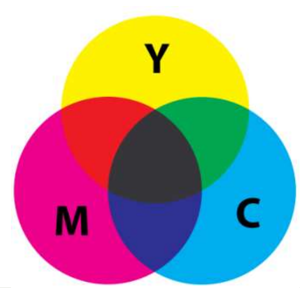
  C + M + Y = Brown?

  C + M + Y = Black (in theory)

  C + M + Y = Gray (in practice)



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



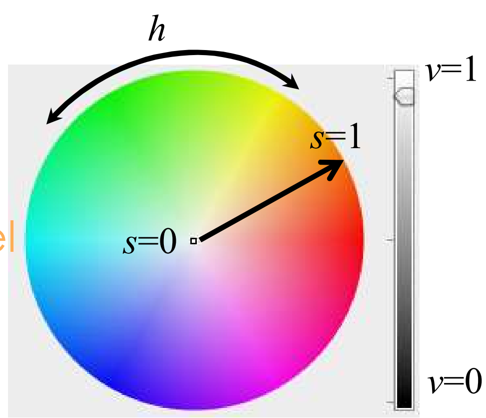
HSV Color Space

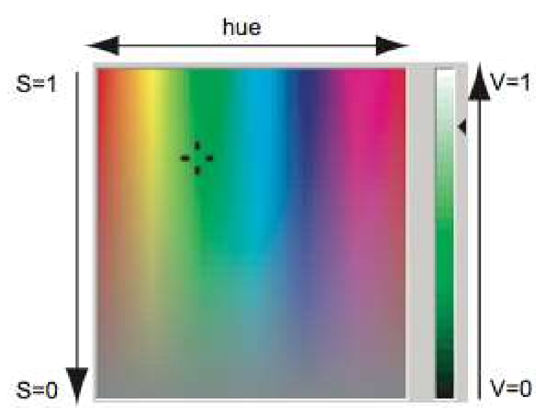
- three floating-point components in $[0,1]$

$$c = (h, s, v) \in [0,1]^3$$

- hue: tint of the color (red, green, blue, yellow, cyan, magenta, yellow, ..
- saturation: strong color (s=1), grayish color ($0<s<1$) or gray (s=0)
- value: luminance; white (v=1), dark ($0<v<1$), or black (v=0)

color wheel



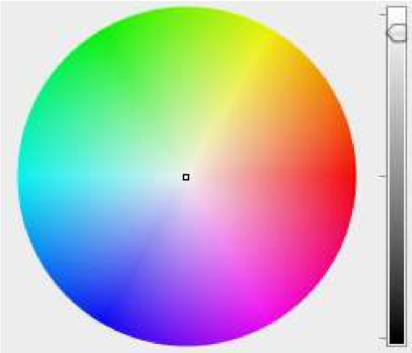


color wheel stretched onto a square

- HSV widgets: typically specify h and s in a 2D canvas and v separately (slider)
- show a 'surface slice' in the RGB cube

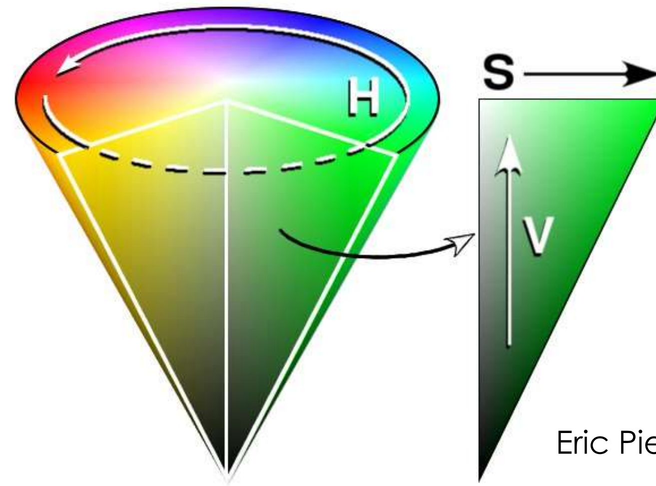
Advantages and Disadvantages

- More intuitive than RGB



- On the other hand it's not perceptually defined
  - Defined in relationship to some RGB space
  - e.g. HSV Saturation poorly models perceived lightness

HSV Color Space



Eric Pierce

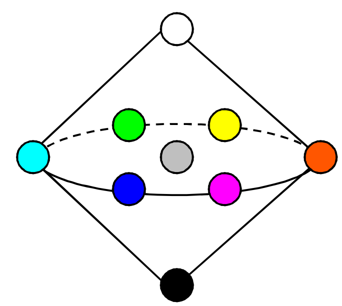
Δ = maxRGB – minRGB

maxRGB = R → H = (G – B)/Δ

maxRGB = G → H = 2+(B – R)/Δ

maxRGB = B → H =4+ (R – G)/Δ

H = (60*H) mod 360



HSV = Hue, Saturation, Value

- 1978, Alvy Ray Smith
- Hue [0,360] is angle about color wheel

  0° = red, 60° = yellow, 120° = green,

  180° = cyan, 240° = blue, 300° = magenta
- Saturation [0,1] is distance from gray

  S = (maxRGB – minRGB)/maxRGB
- Value [0,1] is distance from black

  V = maxRGB

HLS = Hue, Saturation, Lightness

- Double cone, saturation in middle

HSV to RGB and back….

```
void rgb2hsv(float r,float g,float b,
             float& h,float& s,float& v)
{
  float M = max(r,max(g,b));
  float m = min(r,min(g,b));
  float d = M-m;
  v = M;                    //value = max(r,g,b)
  s = (M>0.00001)? d/M:0;   //saturation
  if (s==0) h = 0;          //achromatic case, hue=0 by convention
  else                      //chromatic case
  {
    if (r==M)       h =     (g-b)/d;
    else if (g==M)  h = 2 + (b-r)/d;
    else            h = 4 + (r-g)/d;
    h /= 6;
    if (h<0) h += 1;
  }
}
```

Listing 3.2. Mapping colors from RGB to the HSV space.

```
void hsv2rgb(float r,float g,float b,
             float& h,float& s,float& v)
{
  int   hueCase = (int)(h*6);
  float frac    = 6*h-hueCase;
  float lx      = v*(1 - s);
  float ly      = v*(1 - s*frac);
  float lz      = v*(1 - s*(1 - frac));
  switch (hueCase)
  {
    case 0:
    case 6: r=v;  g=lz; b=lx; break;   // 0<hue<1/6
    case 1: r=ly; g=v;  b=lx; break;   // 1/6<hue<2/6
    case 2: r=lx; g=v;  b=lz; break;   // 2/6<hue<3/6
    case 3: r=lx; g=ly; b=v;  break;   // 3/6<hue/4/6
    case 4: r=lz; g=lx; b=v;  break;   // 4/6<hue<5/6
    case 5: r=v;  g=lx; b=ly; break;   // 5/6<hue<1
  }
}
```

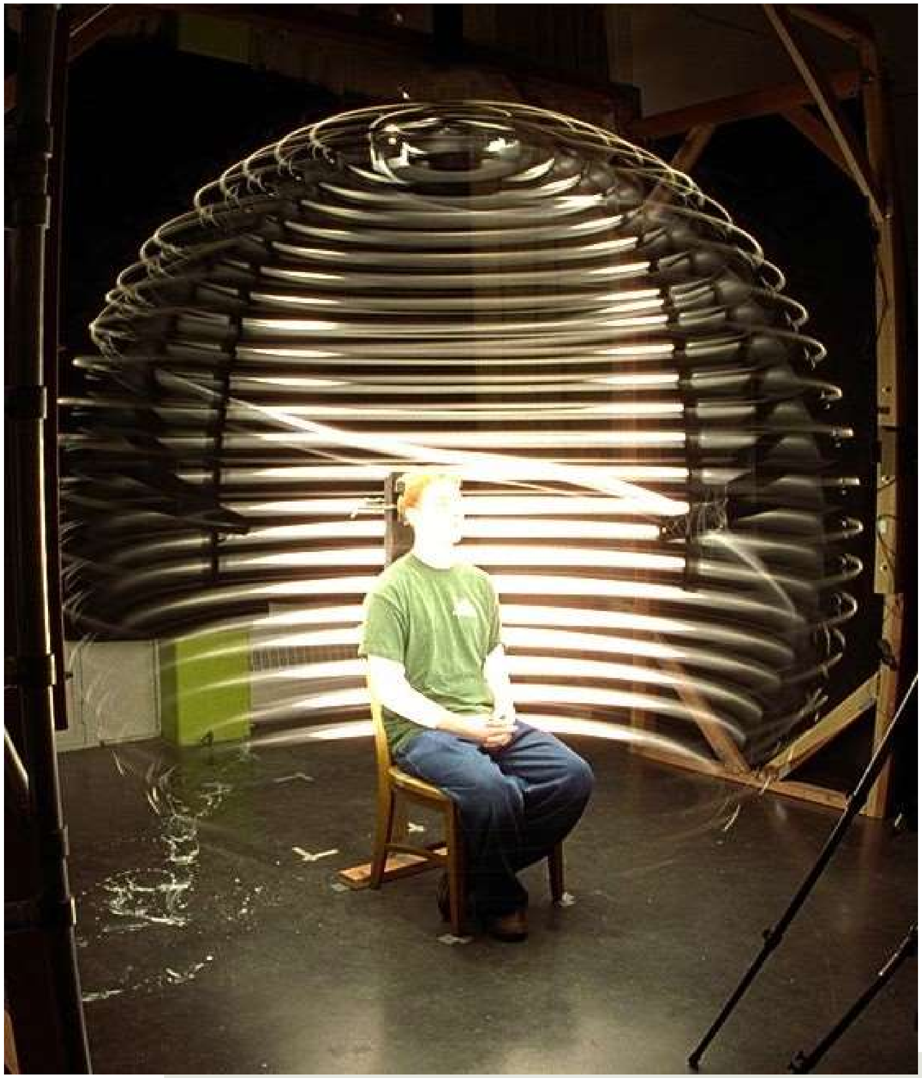
Listing 3.3. Mapping colors from HSV to the RGB space.

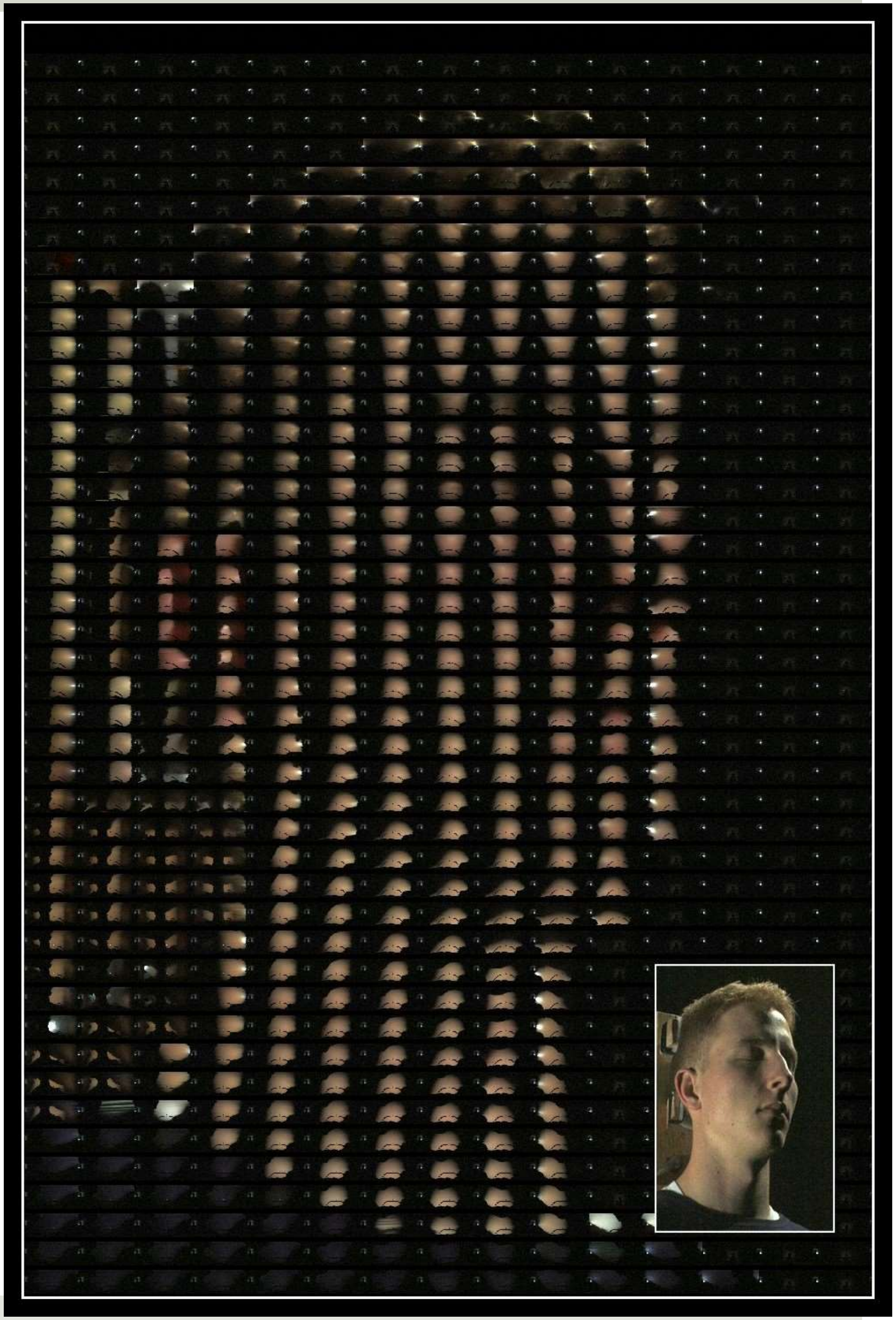
$$R(L_1) + R(L_2) = R(L_1 + L_2)$$

Light Adds

 +  = 

Light Stage





Point Light Sources

Debevec et al., Acquiring the Reflectance Field of a Human Face,
Proc. SIGGRAPH 2000

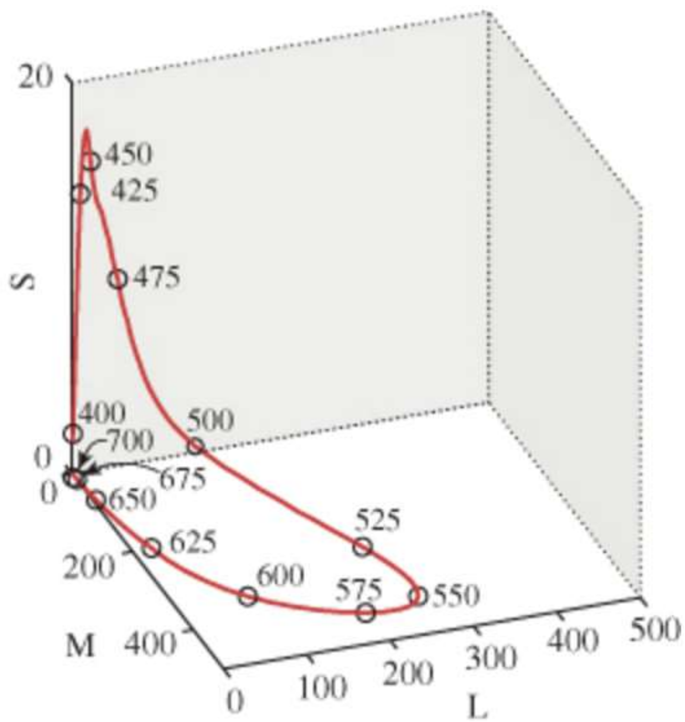
Environment Lighting

Debevec et al., Acquiring the Reflectance Field of a Human Face, Proc. SIGGRAPH 2000

Color Matching Experiments

- Wright and Guild (1920s)
  - Choose lights of 3 different primary colors
  - Show human subject a single-wavelength test light
  - Have subject match test light
    - Use a weighted combination of primaries
    - Weight is luminence
- CIE standard primaries
  - Red (R): 700nm
  - Green (G): 546.1 nm
  - Blue(B): 435.8 nm

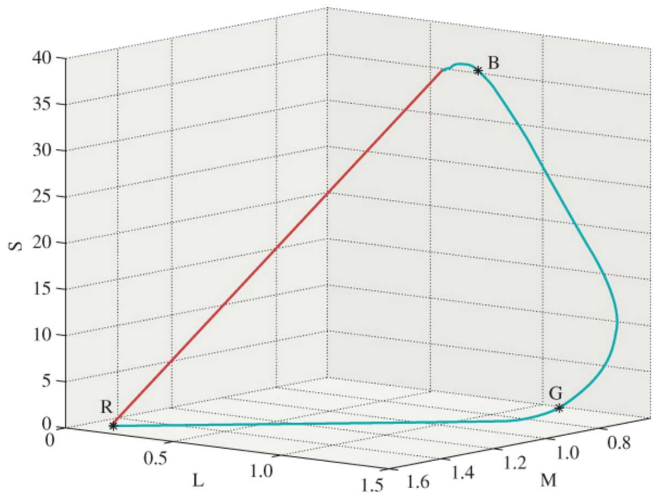
Human Response to Monospectral Light



Points on red curve are wavelengths

Curve position in space shows the response of the the L, M, and S cones

Human Response to Monospectral Light

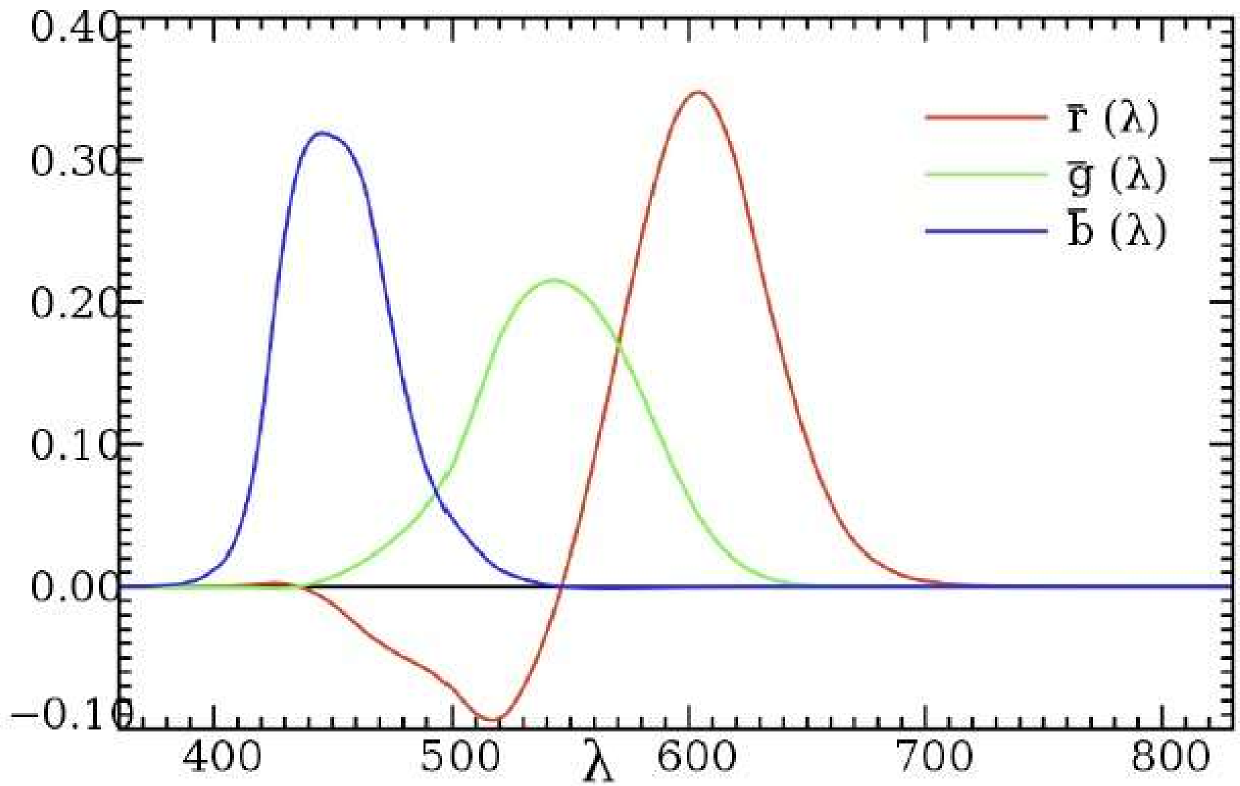


The set of responses to all combinations of monospectral light forms a cone…shaped a little like a horseshoe.

A slice through the cone is shown here.

Can a linear combination of 3 different wavelengths generate all possible color responses?

Color Matching Function for CIE RGB



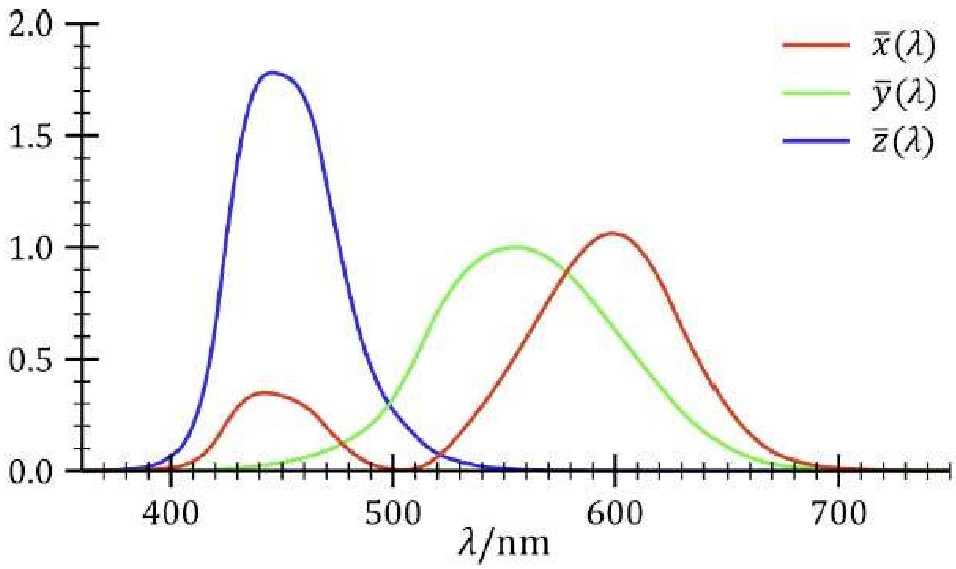
Amounts of the red, green and blue primaries needed to match any color [1]

CIE RGB Color Space

- Experiments by the International Commission on Illumination
  - Commission internationale de l'éclairage
- Defined CIE RGB…you'll notice the negative on the red curve
  - What does this mean happened physically in the experiment?
  - Example: orange = 0.45 R + 0.45 G – 0.1B
  - We can empirically discover that by allowing test subject to add a primary to the test color:

    orange+ 0.1B = 0.45 R + 0.45 G

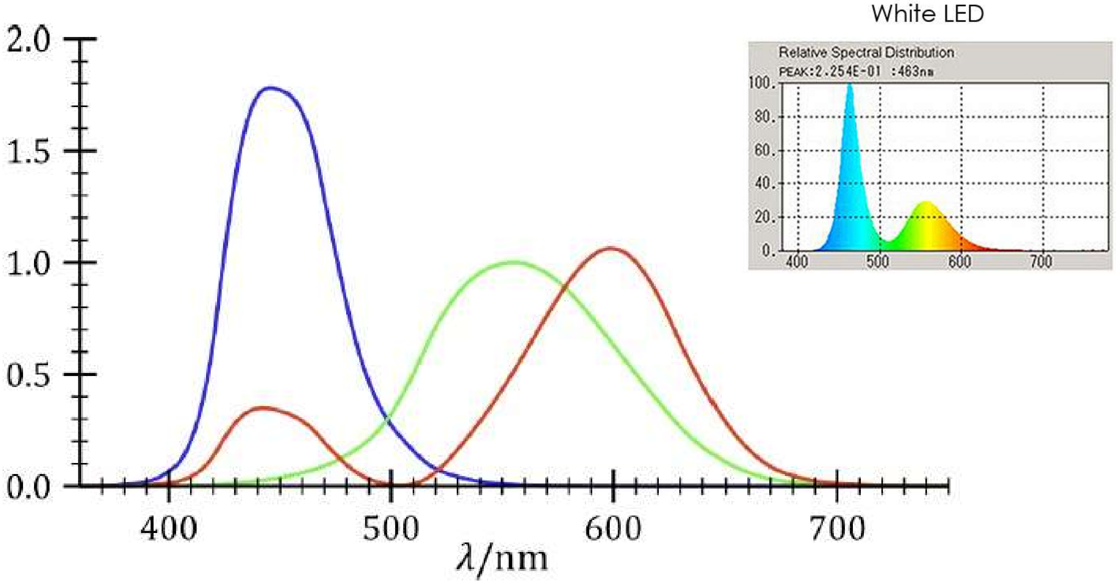
Color Matching Functions for CIE XYZ



Amounts of the XYZ primaries needed to match any color
($\bar{y}$ function is precisely CIE-standardized photopic luminous efficiency, 1931);

- CIE XYZ is another color space based on the experiments…
  - But adjusted to have non-negative functions
- Think of X, Y, and Z being primary colors…but not physically realizable

CIE XYZ Color Space



Amounts of the XYZ primaries needed to match any color
($\bar{y}$ function is precisely CIE-standardized photopic luminous efficiency, 1931):

White LED

←Example Spectrum

A light with spectral power distribution P can be expressed as X**X** + Y**Y** + Z**Z**

where

$$X = k\int P(\lambda)\bar{x}(\lambda)d\lambda$$

$$Y = k\int P(\lambda)\bar{y}(\lambda)d\lambda$$

$$Z = k\int P(\lambda)\bar{z}(\lambda)d\lambda$$

k is 680lmW^{-1}

Integrals are computed numerical…
Functions are a table of values at 1nm intervals
Spectrum sampled at same intervals
Integral becomes a dot product

CIE Color Space

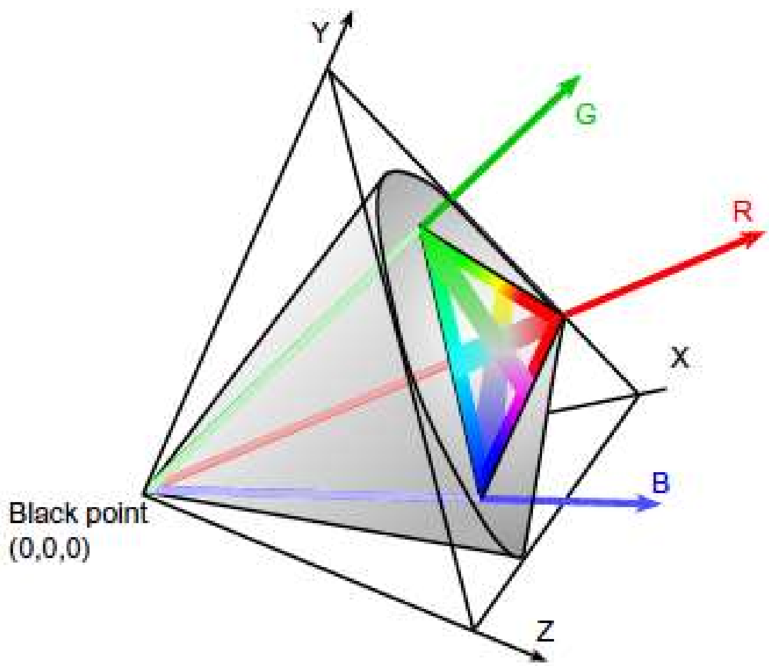


Figure 4.6 The *X*, *Y*, and *Z* axes represent the CIE standard virtual primaries. Within the positive space defined by the axes, the gamut of perceivable colors is represented as a gray solid. The colors that can be created by means of the red, green, and blue monitor primaries are defined by the pyramid enclosed by the *R*, *G*, and *B* lines.

From *Information Visualization* by Colin Ware

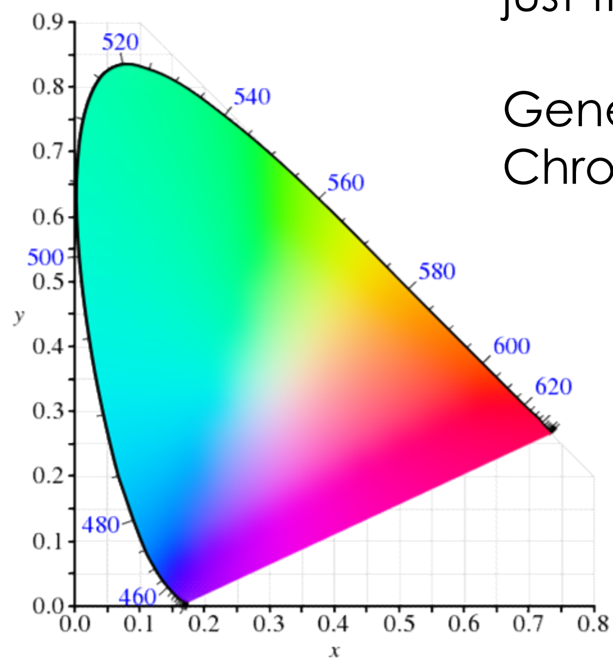
xyY: Separates Chromaticity and Luminence

Formed from X,Y,Z expression of a color

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$



Note: $x + y + z = 1$

Used to specify intensity independent colors using just the x,y coordinates

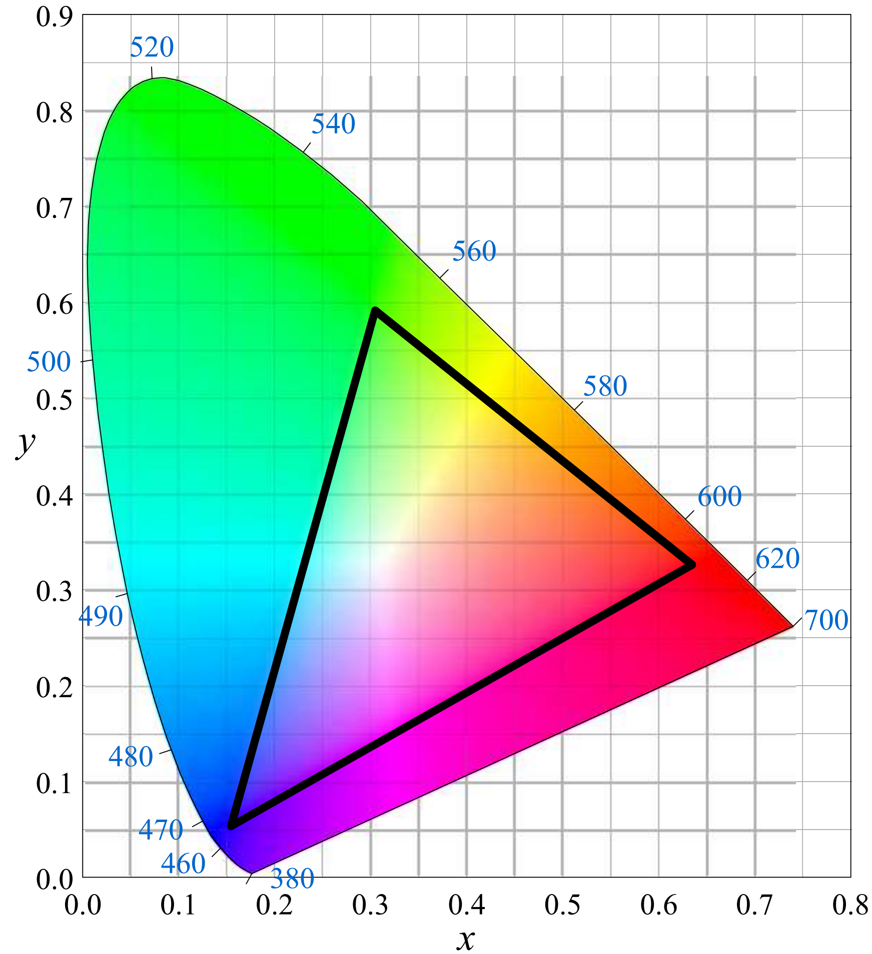
Generates the CIE Chromaticity diagram

CIE Chromaticity Diagram

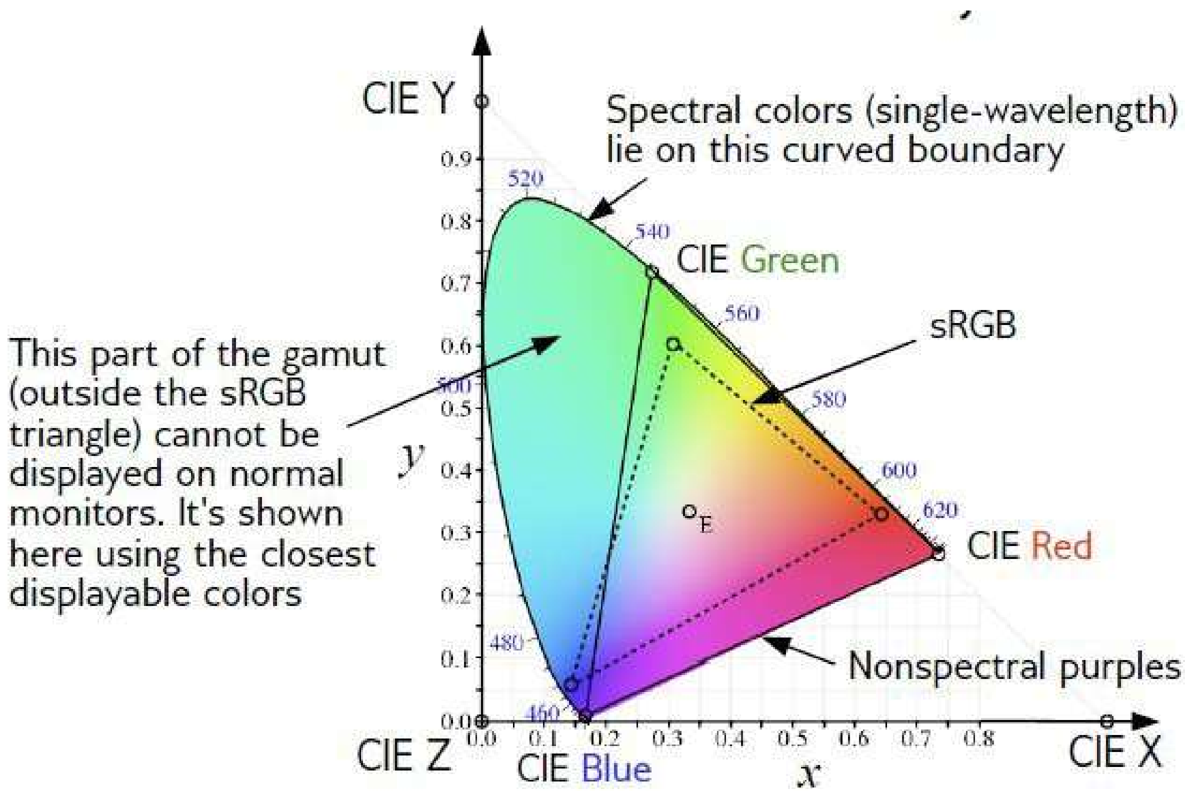
- What runs around the edge of the horseshoe?
- What is inside the horseshoe?
- **Gamut:** Portion of the spectrum reproduced by a given color space

  Any guess as to what the triangle represents?
- **Quick Quiz:**
  Are the colors shown inside the diagram correct?

  Why or why not?



CIE XYZ Gamut



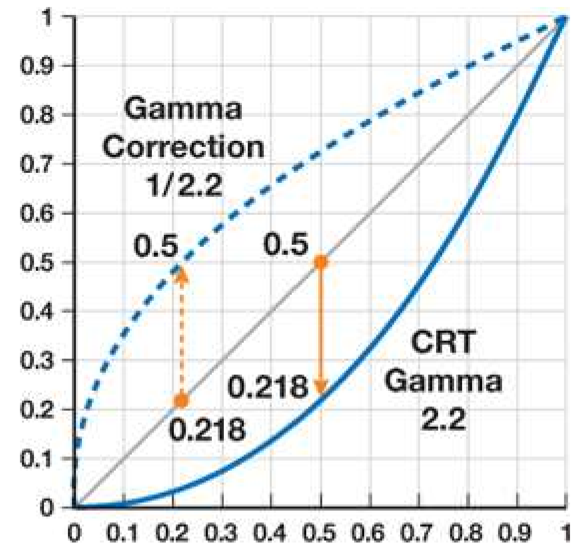
All visible chromaticities mapped to xy plane

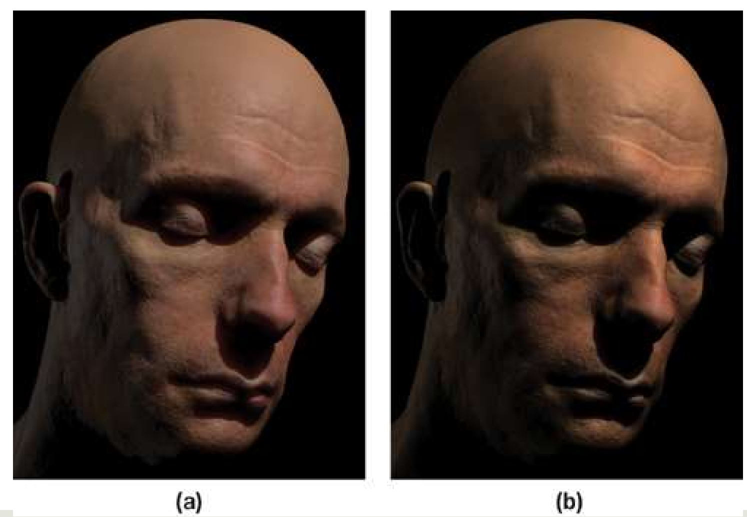
Gamma Correction

- We perceive differences in intensity more carefully for darker shades
- Monitors accommodate this feature

$$I = V^{\gamma}$$

- Gamma usually between 2 and 2.5





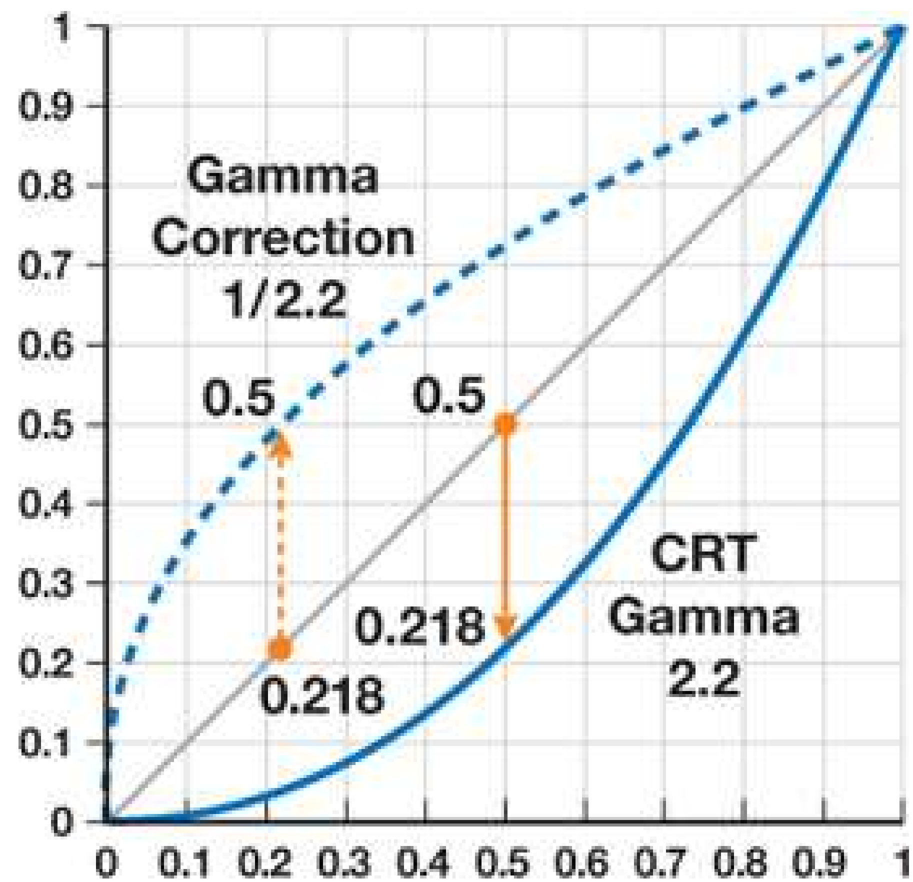
Figures from The Importance of Being Linear
Larry Gritz and Eugene d'Eon
NVIDIA Corporation

Gamma Correction Example

- A red value of 0.5 is between ¼ and 1/5 as bright as a red value of 1.0
- If we transform:

$$(0.5)^{\frac{1}{2.2}} = 0.729$$

we get a red value half as bright as 1.0 when displayed.



Design Tip: Use High Saturation for Small Symbols

Figure 4.9 (a) Large samples of saturated colors. (b) Large samples of the same colors less saturated. (c) Small samples of the same saturated colors. (d) Small samples of the less saturated colors.

[G4.1] Use more saturated colors when color coding small symbols, thin lines, or other small areas. Use less saturated colors for coding large areas.

Cultural Notions of Color

- Cultural evidence implies some colors more perceptually important

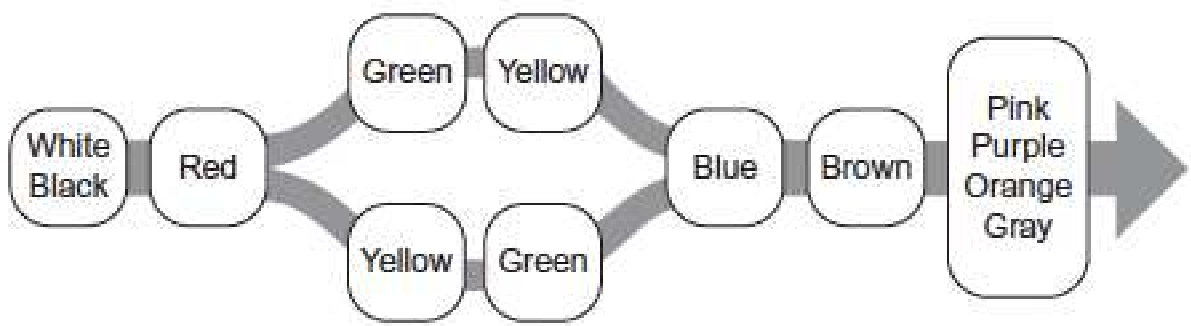
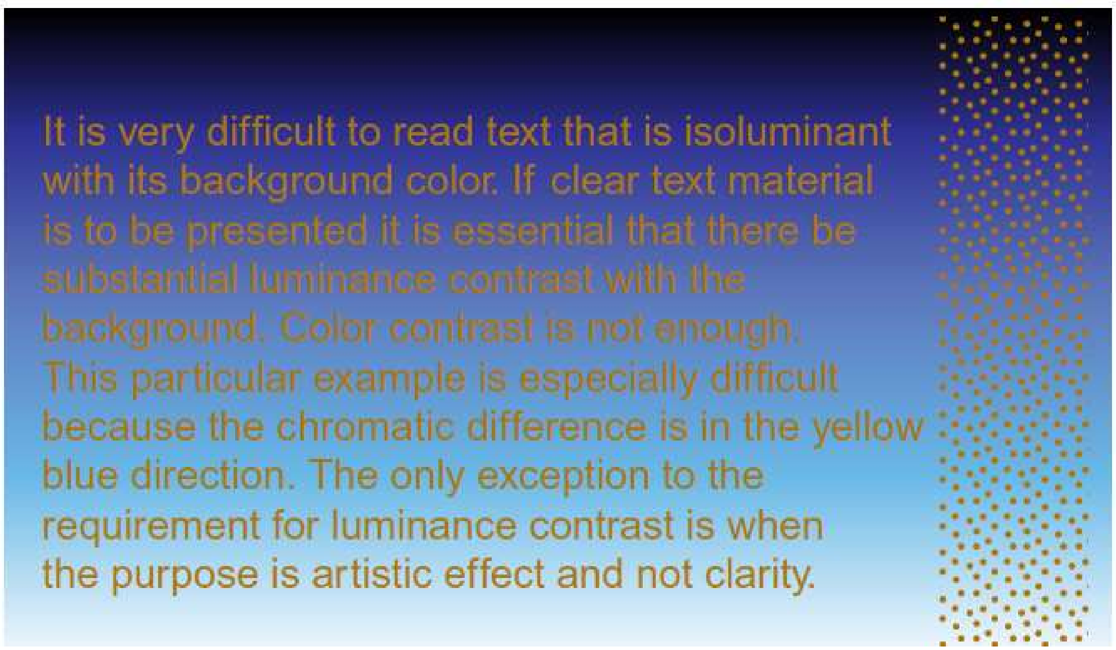
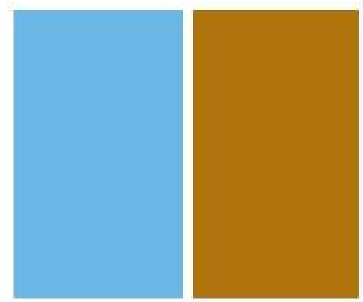


Figure 4.11 This is the order of appearance of color names in languages around the world, according to the research of Berlin and Kay (1969). The order is fixed, with the exception that sometimes yellow is present before green and sometimes the reverse is the case.

Design Tip:
Use Luminance Contrast for Text

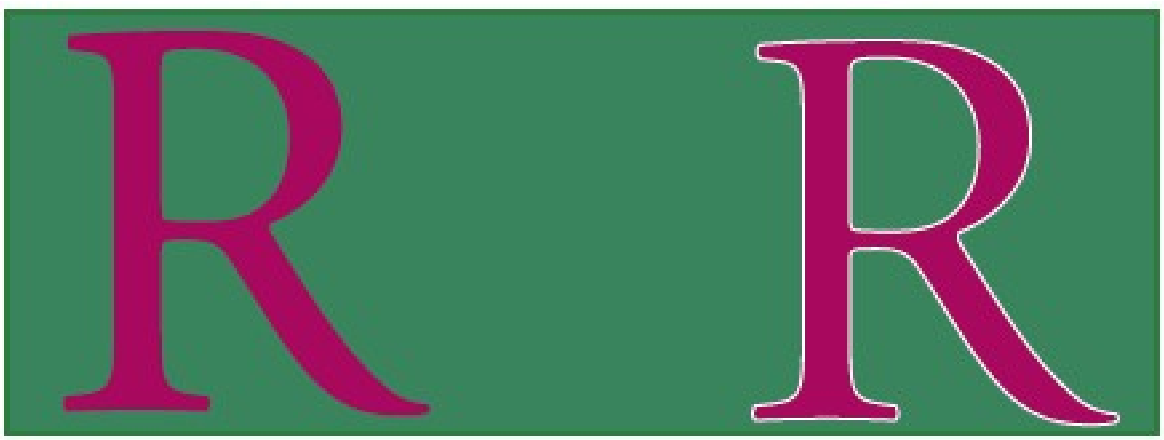
- Perceptually, isoluminant color changes carry only 1/3 the info that a change in luminance does
  - e.g. varying from blue to yellow versus black to white



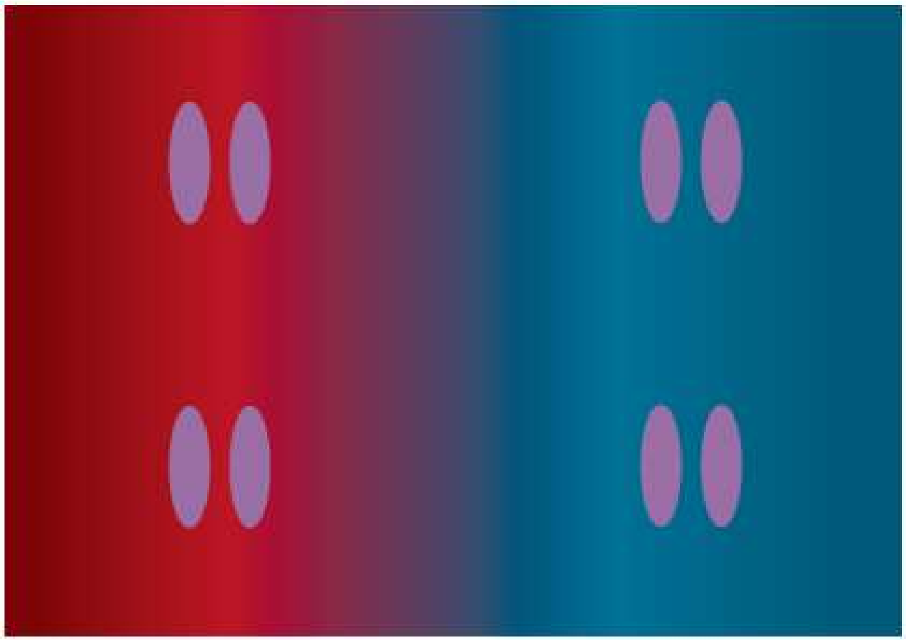


Luminance Contrast

- Important for
  - Visualizing fine-grained changes in data
  - In perceiving depth
  - Perception of motion
  - …and displaying text

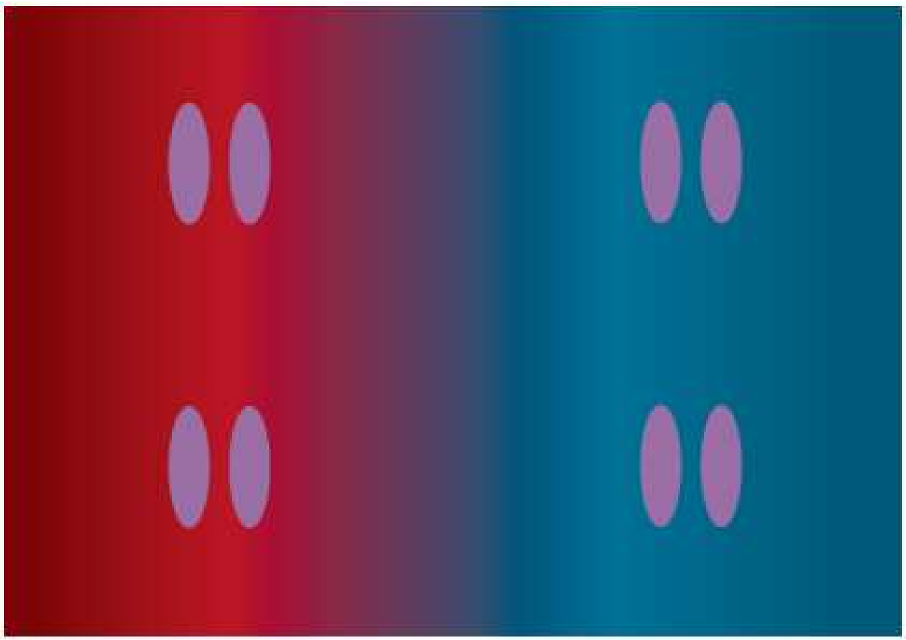


Color Contrast



Are all the ellipses the same color?

Color Contrast



When designing a visualization be aware that large color context areas
change perception of color.

People will not perceive the ellipses as identical

A Color Specification Interface

- From *Information Visualization* by Ware

[G4.8] In an interface for specifying colors, consider laying out the red–green and yellow–blue channel information on a plane. Use a separate control for specifying the dark–light dimension.

[G4.9] In an interface for designing visualization color schemes, consider providing a method for showing colors against different backgrounds.

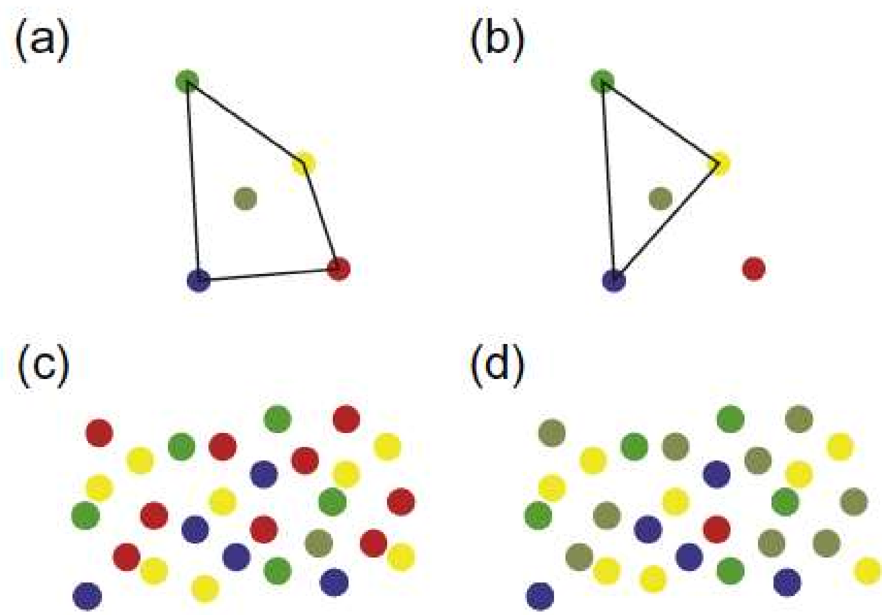
Visualizing Categorical or Nominal Data

- "One reason chromaticity is preferred for nominal data is that the alternatives are generally worse"
- Any guesses how many greyscale values people can successfully distinguish in memory after seeing it used to label nominal data?

[G4.11] Consider using red, green, yellow, and blue to color code small symbols.

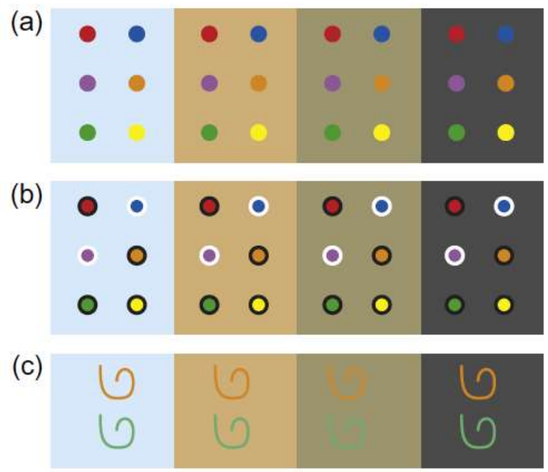
What if 4 colors does not suffice?

- Keep adding colors from outside the convex hull of existing colors (in CIE space)



More Nominal Data Advice

[G4.13] If colored symbols may be nearly isoluminant against parts of the background, add a border having a highly contrasting luminance value to the color, for example, black around a yellow symbol or white around a dark blue symbol.

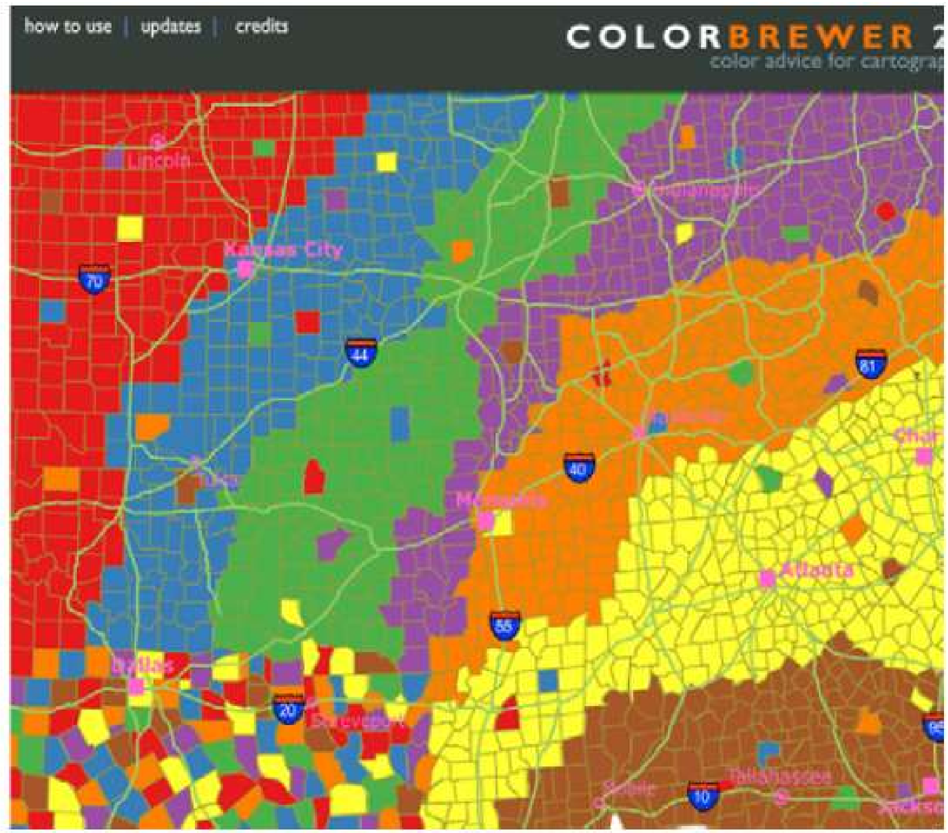


More Nominal Data Advice

[G4.14] To create a set of symbol colors that can be distinguished by most color-blind individuals, ensure variation in the yellow–blue direction.

[G4.15] Do not use more than ten colors for coding symbols if reliable identification is required, especially if the symbols are to be used against a variety of backgrounds.

Which do you prefer?



Use Low Saturation for Larger Areas

[G4.16] Use low-saturation colors to color code large areas. Generally, light colors will be best because there is more room in color space in the high-lightness region than in the low-lightness region.

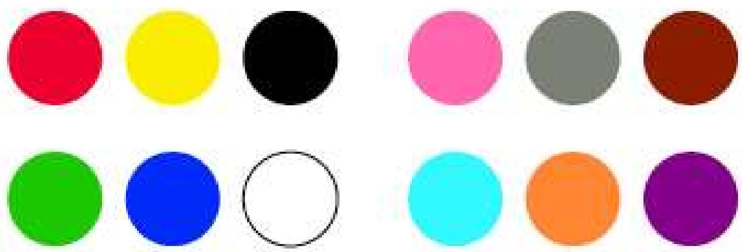
[G4.17] When color coding large background areas overlaid with small colored symbols, consider using all low-saturation, high-value (pastel) colors for the background, together with high-saturation symbols on the foreground.

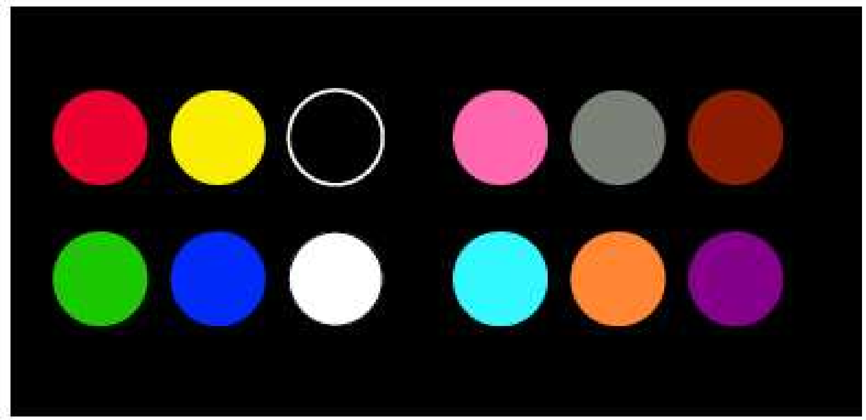
Be Aware of Cultural Conventions

Conventions. Color-coding conventions must sometimes be taken into account. Some common conventions are red = hot, red = danger, blue = cold, green = life, and green = go. It is important to keep in mind, however, that these conventions do not necessarily cross cultural borders. In China, for example, red means life and good fortune, and green sometimes means death.

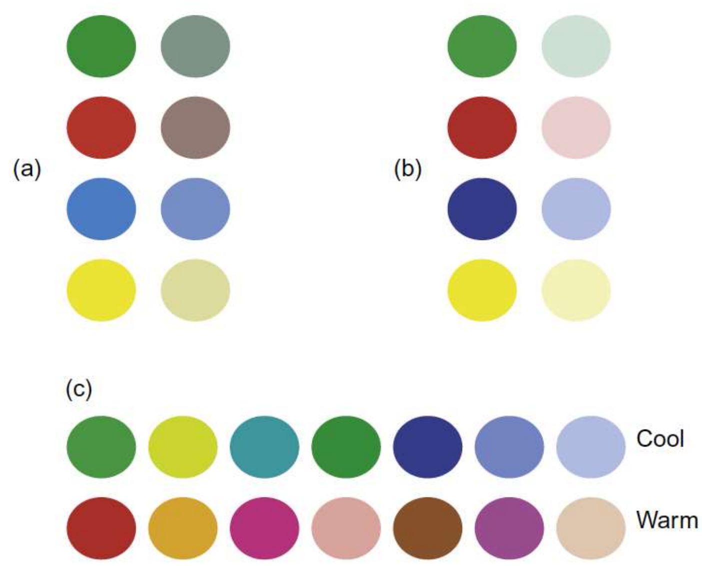
From *Information Visualization* by Colin Ware

Suggested Colors for Labeling





Color Families



"Interior designers often consider a family of warm colors (nearer to red in color space) to be distinct from a family of cool colors (nearer to blue and green in color space), although the psychological validity of this is questionable." – Colin Ware

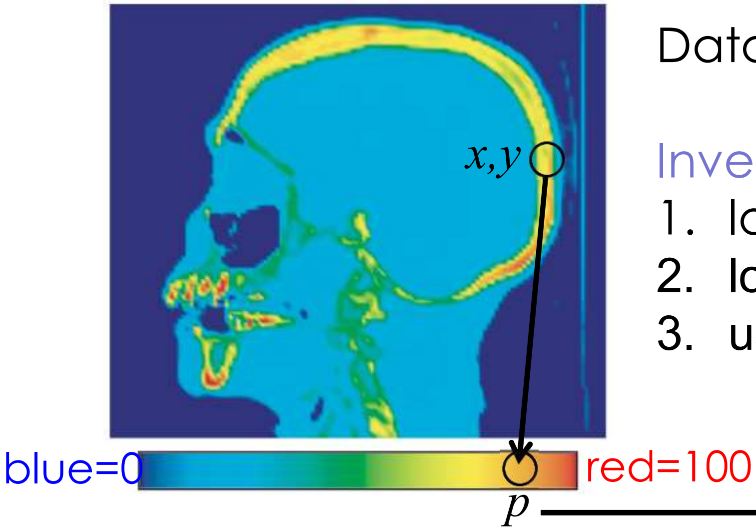
Coloring Continuous Data

- Coloring to denote continuous data is called pseudo-coloring
  - Such a mapping from value to color is called a choropleth
- Anyone know the standard weather map colors?
- Anyone know the standard elevation map colors in geography?
- What is the most used colormap?
  - Is it any good?

Coloring Continuous Data

- Coloring to denote continuous data is called pseudo-coloring
  - Such a mapping from value to color is called a choropleth
- Anyone know the standard weather map colors?
  - blue-cyan-green-yellow-orange-red
- How about elevation in geography?
  - blue-green-brown-white
- What is the most used colormap?
  - Rainbow
  - Is it any good?
    - Probably not

Keyed Lookup Example



Data values mapped to RGB colors via a colormap

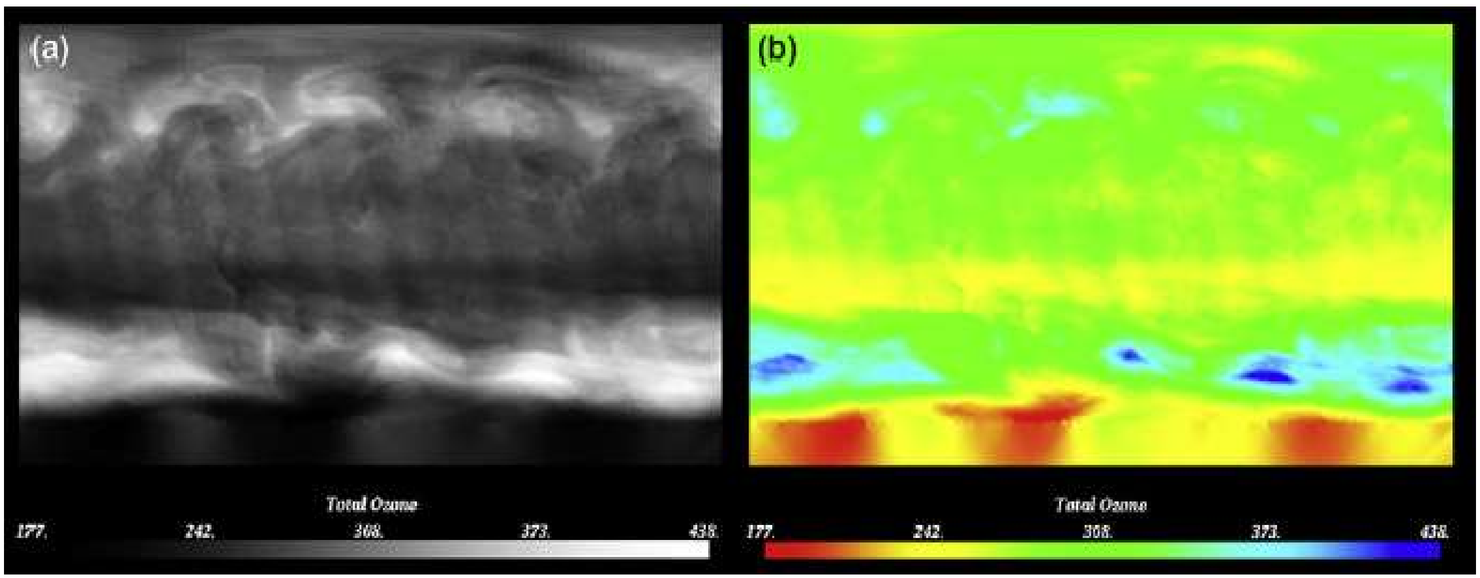
Invert mapping:

1. look at some point (x,y) in the image $\rightarrow$ color c
2. locate c in colormap at some position p
3. use the colormap legend to derive data value s from p

answer: $s = 90$

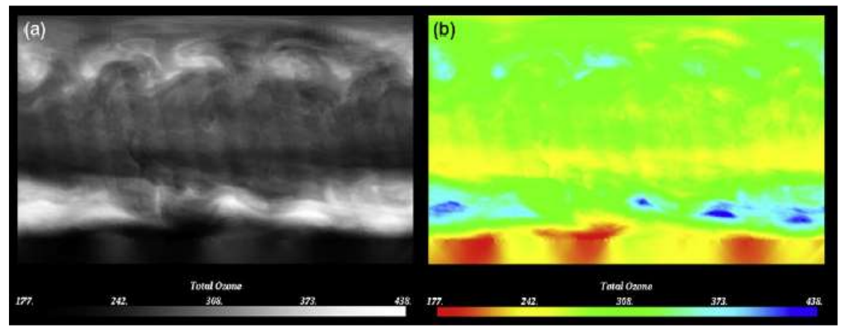
Rainbow versus Black and White

- People intuitively order greyscale by increaisng luminance
- Not so with the rainbow colors
  - No commonly used ordering in studies…



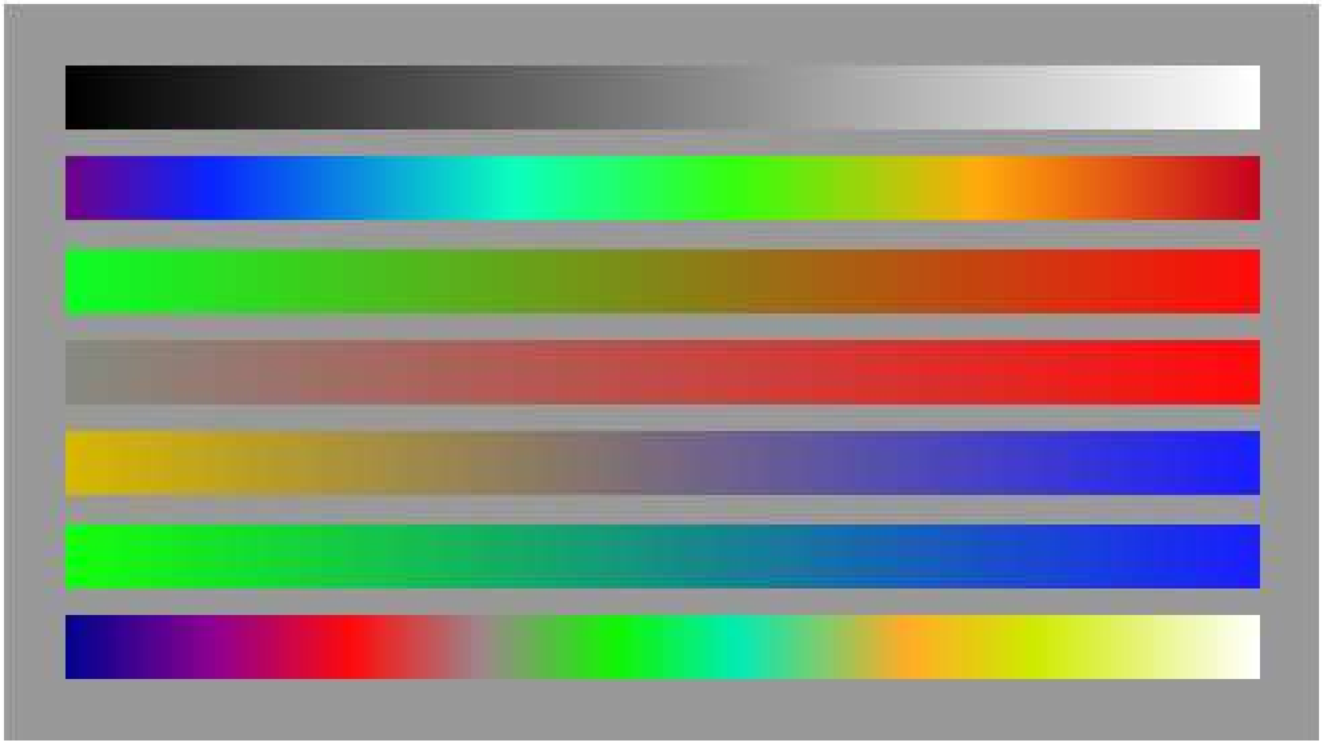
Rainbow versus Black and White

- Greyscale is superior at delineating structure and form in data



- Rainbow
  - Reflects common idiom in meteorology
  - Reduces errors in reading from keys
  - Enables better estimation of quantity at a point
    - Why?

Common colormaps



Greyscale

Rainbow

Red-Green

Saturation

2 for potential colorblindness

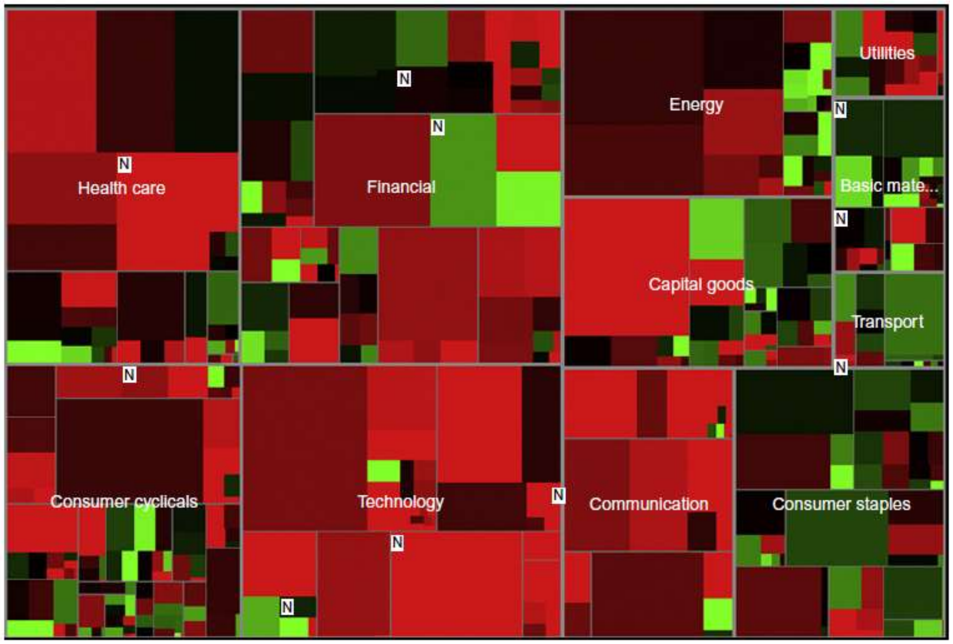
Increasing lightness

Pseudo-color Tip

[G4.20] If it is important to see highs, lows, and other patterns at a glance, use a pseudocolor sequence that monotonically increases or decreases in luminance. If reading values from a key is also important, cycle through a variety of hues while trending upward or downward in luminance.

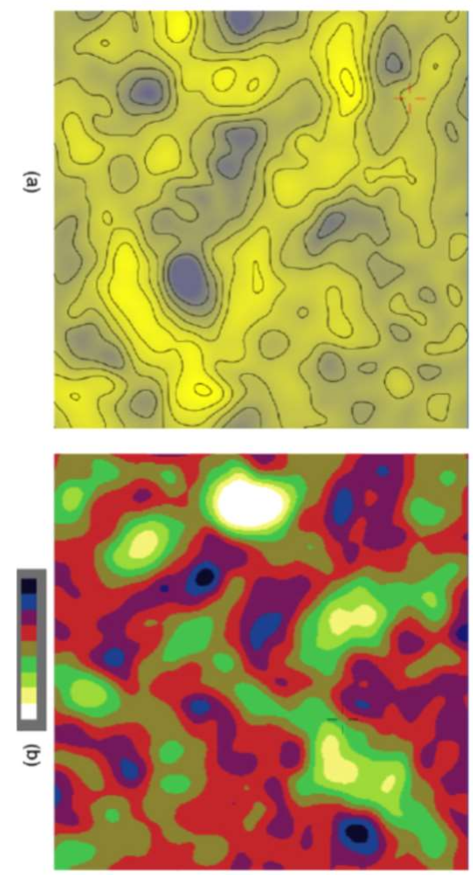
Ratio Sequences

- Ratio sequence is an interval sequence including 0
- Colormap should strongly delineate 0 point, positive and negative

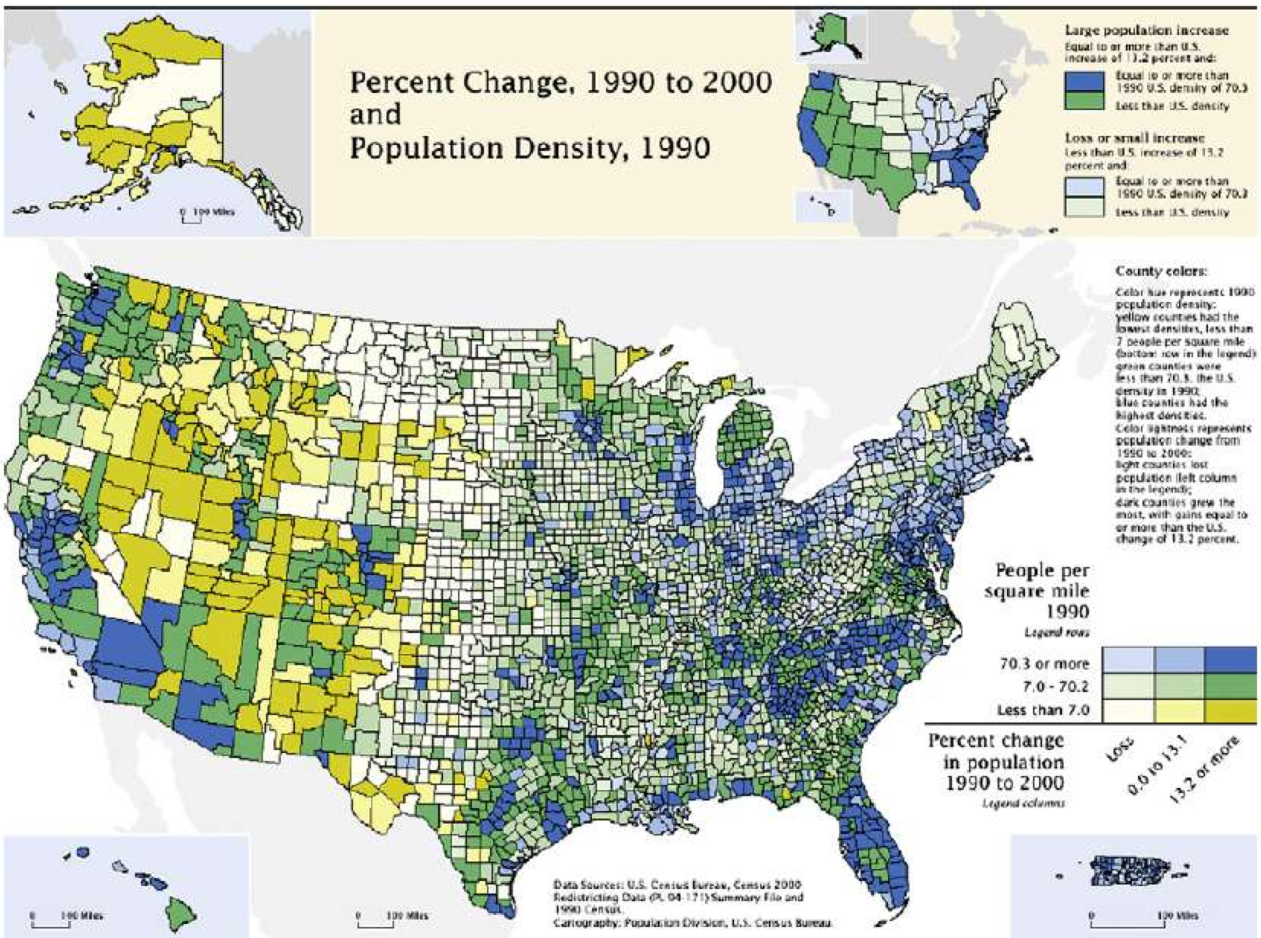


Contours and Interval Colormaps

- For estimating value from color
  - Isocontours help
  - Stepped-functions show better user performance



Bivariate Colormaps



- How are the 2 variables shown here?

Colormap Math

Basic idea

- Map each scalar value $f \in \mathbf{R}$ at a point to a color via a function $c : [0,1] \rightarrow [0,1]^3$

Color tables

- precompute (sample) c and save results into a table $\{c_i\}_{i=1..N}$
- index table by normalized scalar values

input data | scalar value f

determine input range | scalar value range $[f_{\min}, f_{\max}]$

normalize input to [0,N] | $i = N \dfrac{f - f_{\min}}{f_{\max} - f_{\min}}$

Color mapping ← $c_i = c\left(\dfrac{i}{N}\right)$ ← $c : [0,1] \rightarrow [0,1]^3$

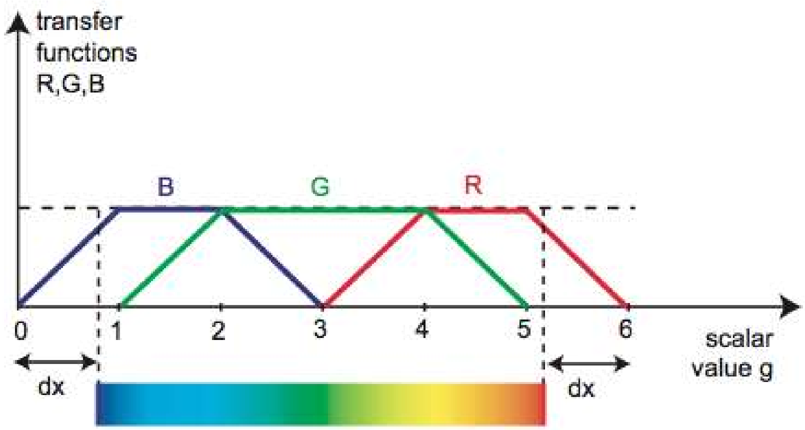
color table

desired color transfer function

f is color-mapped to c_i

Implementing the Rainbow Map

- probably the most (in)famous in data visualization
- intuitive ‘heat map’ meaning
    - cold colors = low values
      warm colors = high values

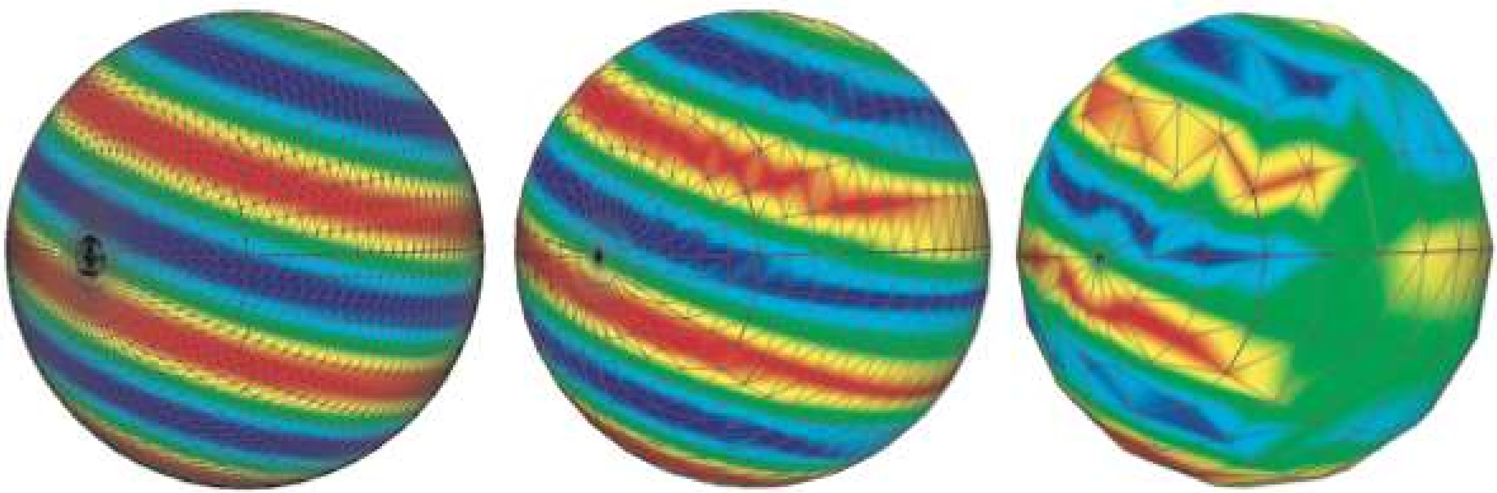



```
void c(float f, float& R, float& G, float& B)
{
    const float dx = 0.8;
    f = (f<0)? 0 : (f>1)? 1 : f;          //clamp f in [0,1]
    g = (6-2*dx)*f + dx;                  //scale f to [dx,6-dx]
    R = max(0,(3-fabs(g-4)-fabs(g-5))/2);
    G = max(0,(4-fabs(g-2)-fabs(g-4))/2);
    B = max(0,(3-fabs(g-1)-fabs(g-2))/2);
}
```

Ideally you should render the colors per-pixel

Where to apply the colormap?

- per pixel drawn – better results than per-vertex colormapping
- done using 1D textures



64x64 points   32x32 points   16x16 points

Explanation

- per-vertex: $f \to c(f) \to$ interpolation$(c(f))$ color interpolation can fall outside colormap!
- per-pixel: $f \to$ interpolation$(f) \to c($interpolation$(f))$ colors always stay in colormap