CSE528 Computer Graphics: Theory, Algorithms, and Applications

Hong Qin
Rm. 151, NEW CS
Department of Computer Science
Stony Brook University (SUNY at Stony Brook)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.sunysb.edu; or qin@cs.stonybrook.edu;
http://www.cs.stonybrook.edu/~qin

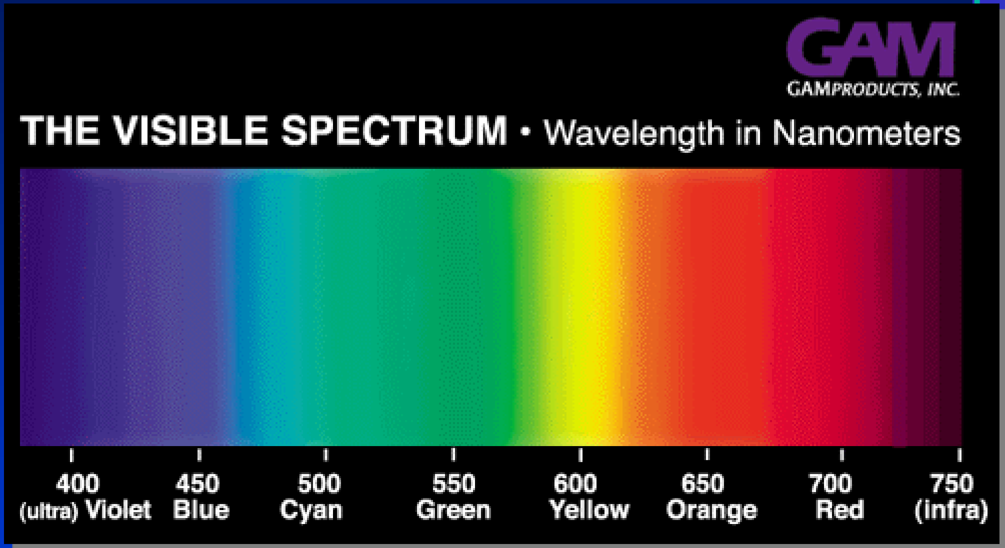
Color Theory and Human Visual System

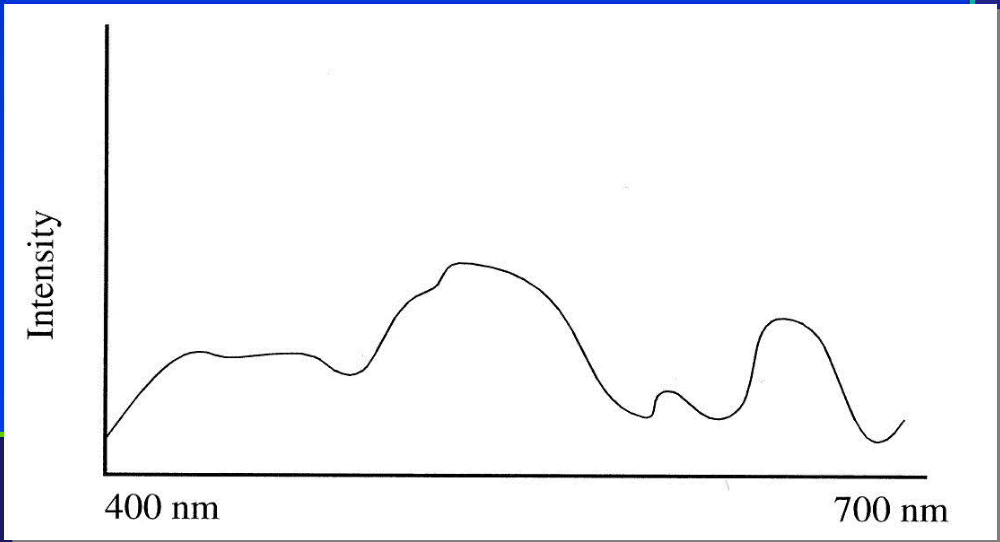
Optics

- The study of light has 3 sub-fields
  - *Geometric optics*: study of the particle nature of light
  - *Physical optics*: study of the wave nature of light
  - *Quantum optics*: study of the dual wave-particle nature of light and attempt to construct unified theories to support duality. Wave "packets" called photons
- Computer graphics most concerned with geometric optics (but need some of the others, too)

Color

- Visible spectrum wavelengths 400-700 nm
- A given color has some distribution of these wavelengths
- Intensity of each wavelength determines contribution to color





Color

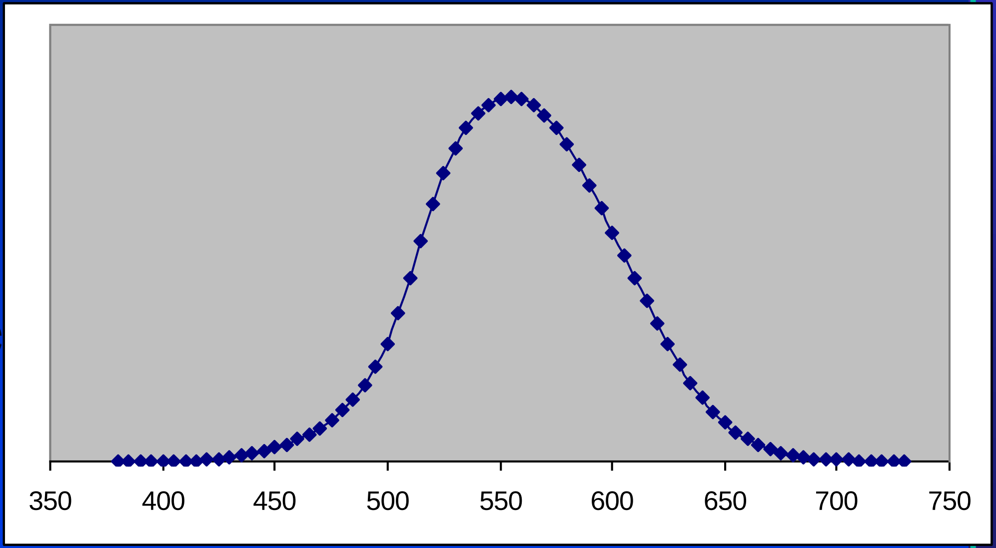
- Color is determined by the wavelength of visible light
- Still use geometric optics
- But need to account for wavelength in reflectance (BRDF) and index of refraction
- What natural phenomena can you think of that are wavelength dependent?

Wavelength Sampling

- We could try to compute image for every possible wavelength and then combine
  - Would take forever
- Sample a representative set of wavelengths
  - How many samples?
  - Where?

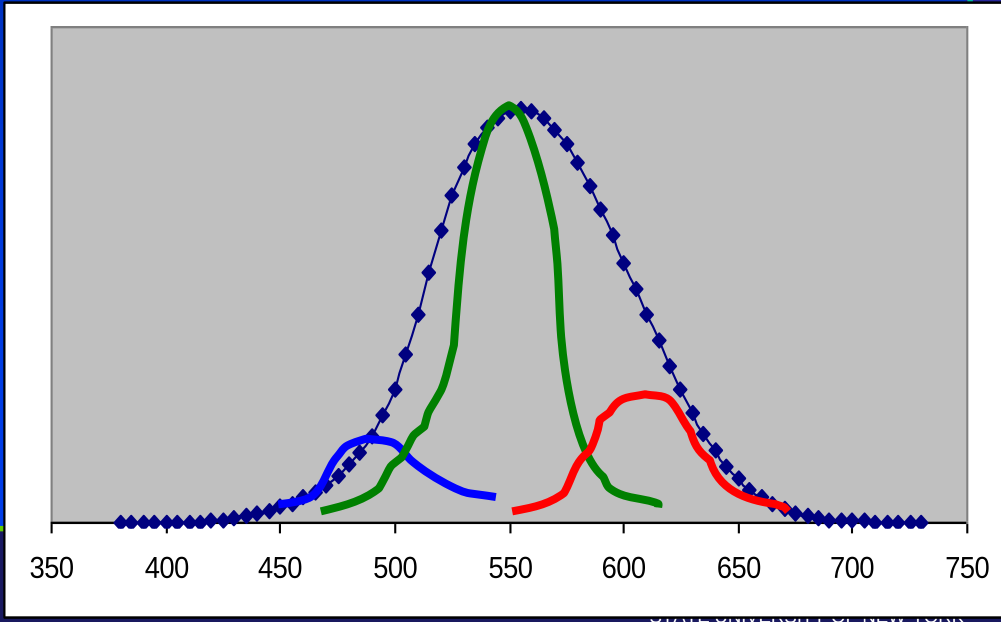
Where to Sample?

- Photometry tells us that some wavelengths are more important than others to human perception.
  - Human response curve looks something like this:



Where to Sample?

- So, pick a few samples wavelengths.
  - Compute an image for each.
  - Reconstruct with basis functions.
  - Weight of each sample determined by human response curve.
  - (Also need color space transformations)



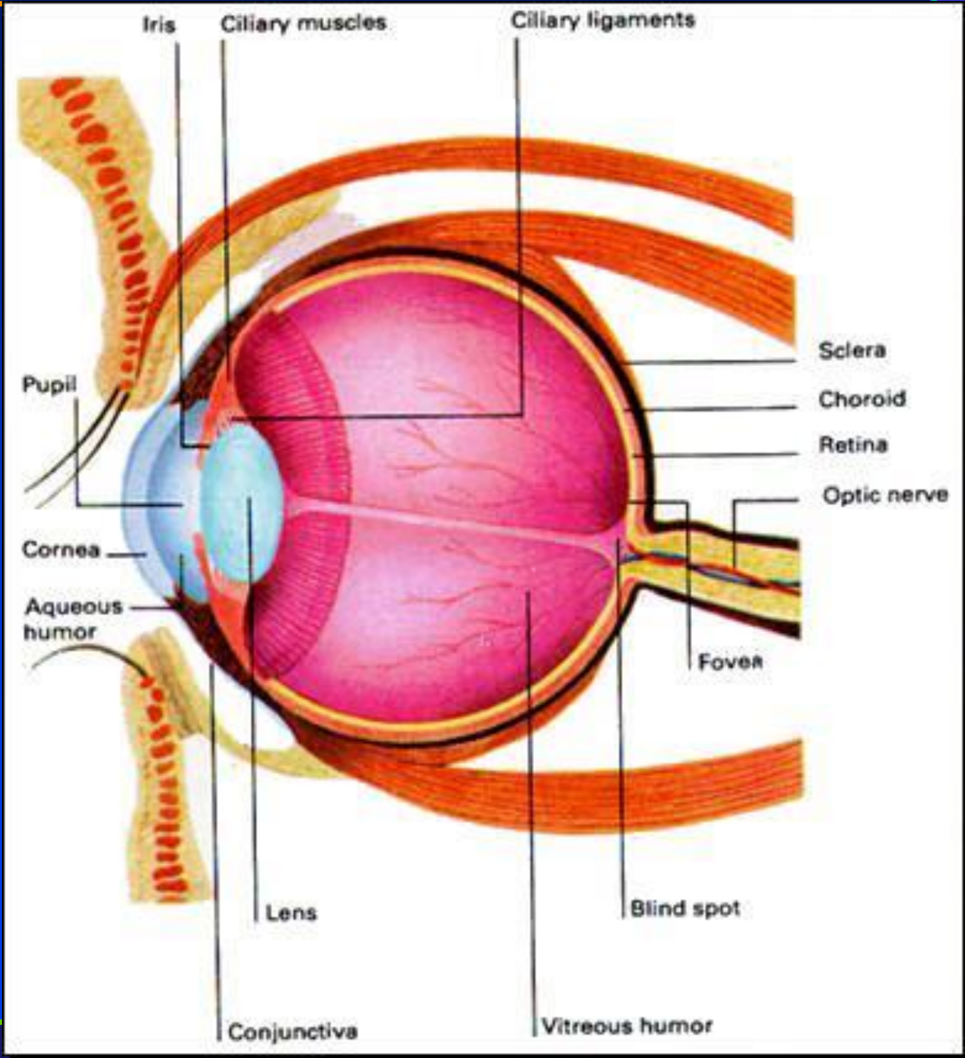
Human Visual System and Color Theory

- Today: human visual system
- Human eye
- Color models
- Color perception

Human Perception (Visual) System

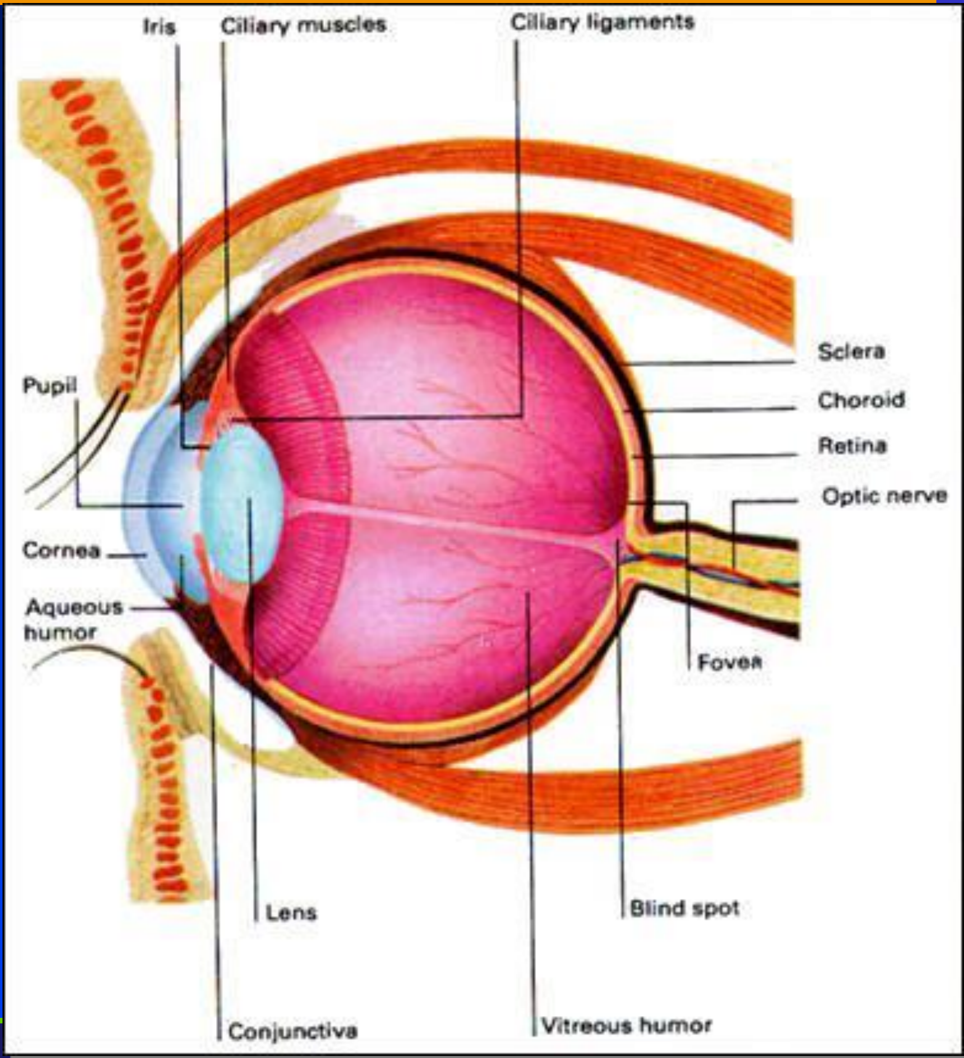
How do we perceive the visible world?

1. Light enters eye and strikes lens
2. Muscles expand and contract to focus light on retina at the back of the eye



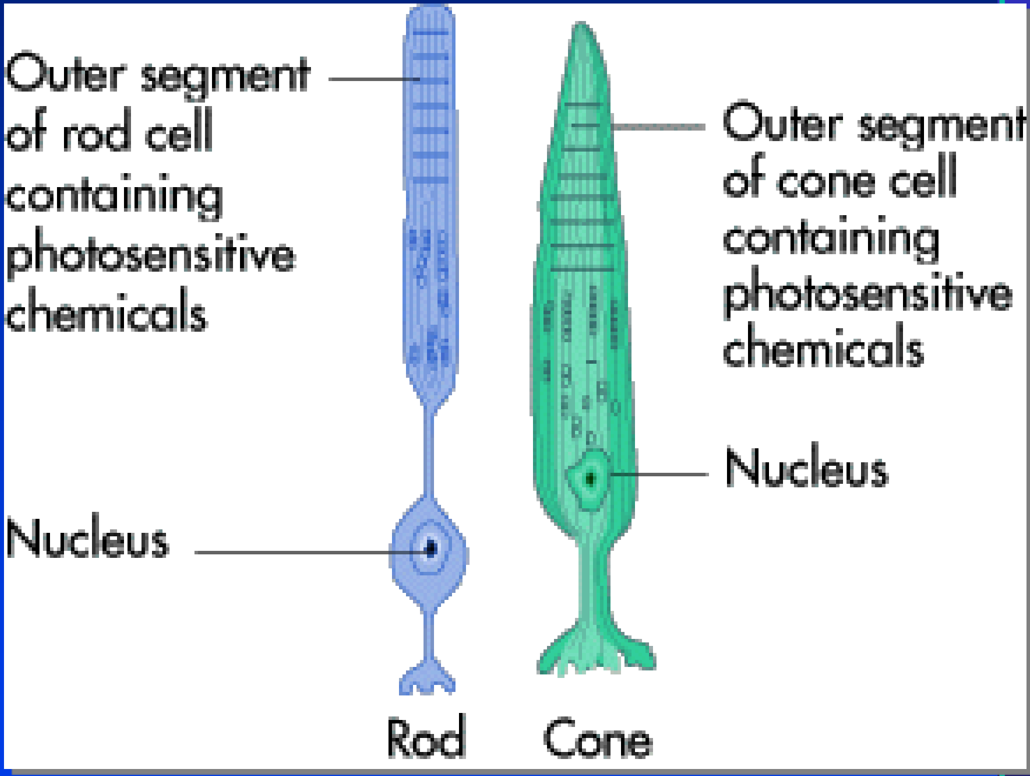
Human Perception (Visual) System

3. Retina senses light and contains **cone cells** and **rod cells**
4. Retinal nerve fibers connect to optic nerve, which carries signals to brain, where they are interpreted as an image



Rods

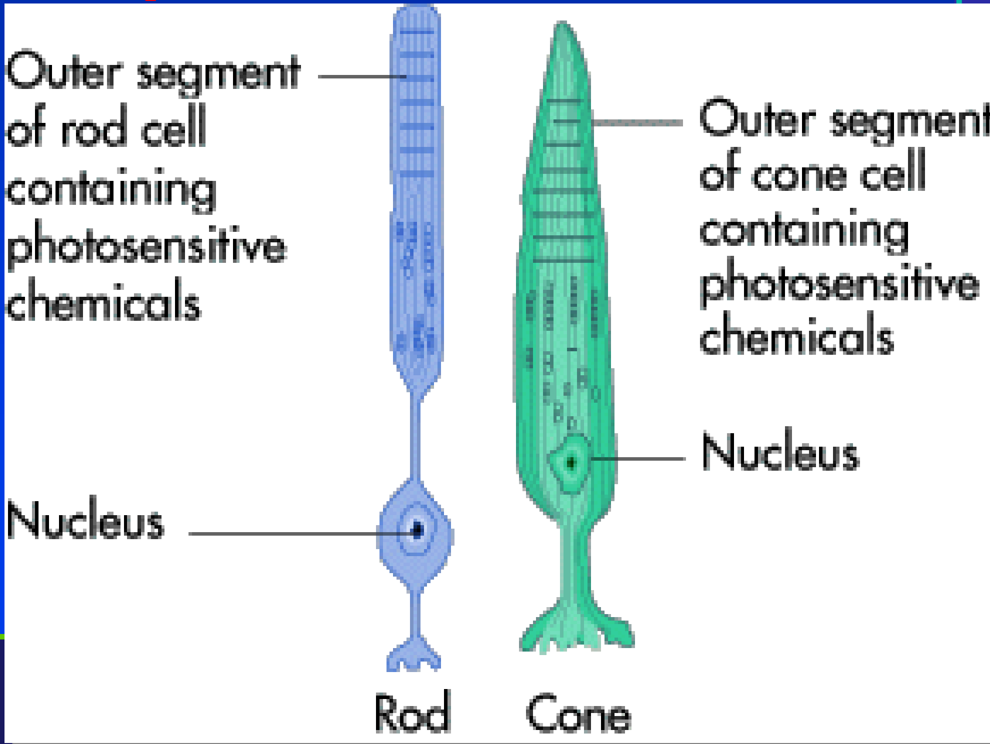
- Spread all over retina
- 75-150 million
- Low resolution
- Do not detect color
- Very sensitive to low light
- This is called **scotopic vision**



Cones

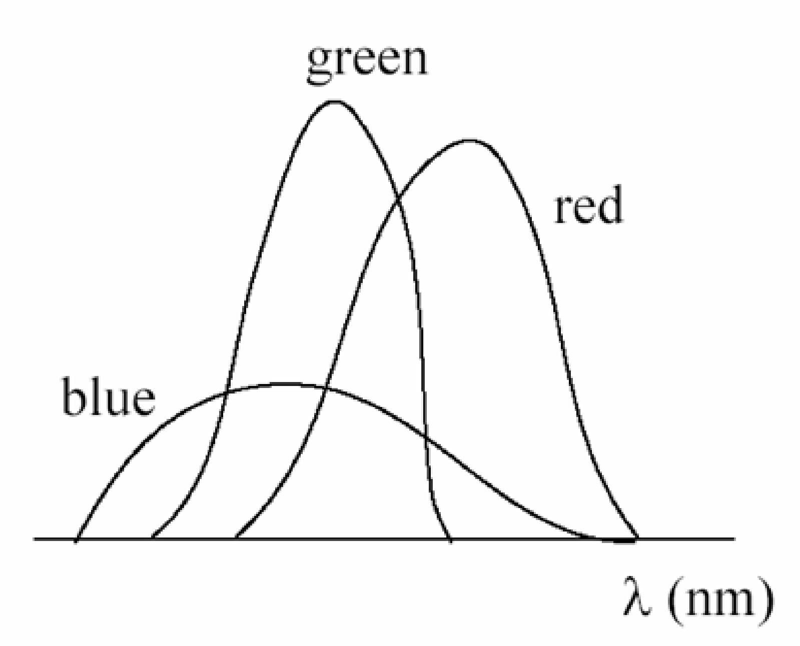
- A dense array of cells around central portion of retina
- 6-7 million total
- High-resolution, detect color
- Require bright color (This is called **photopic vision)**

- Our eyes, therefore, have a fixed resolution, much like a computer screen
- Our brains permits us to perceive the images as continuous even though they really aren't
- This happens through a process known as *interpolation*



Cones

- Detect color on the retinal surface
- Can be divided into “red” cones (~60%), “green” cones (~30%) and “blue” cones (~10%)
- A complex mixing process takes place inside the brain to generate the final color
- Graph of receptor sensitivities of each type of cone:
- Question: what kinds of colors are we best able to distinguish from each other?

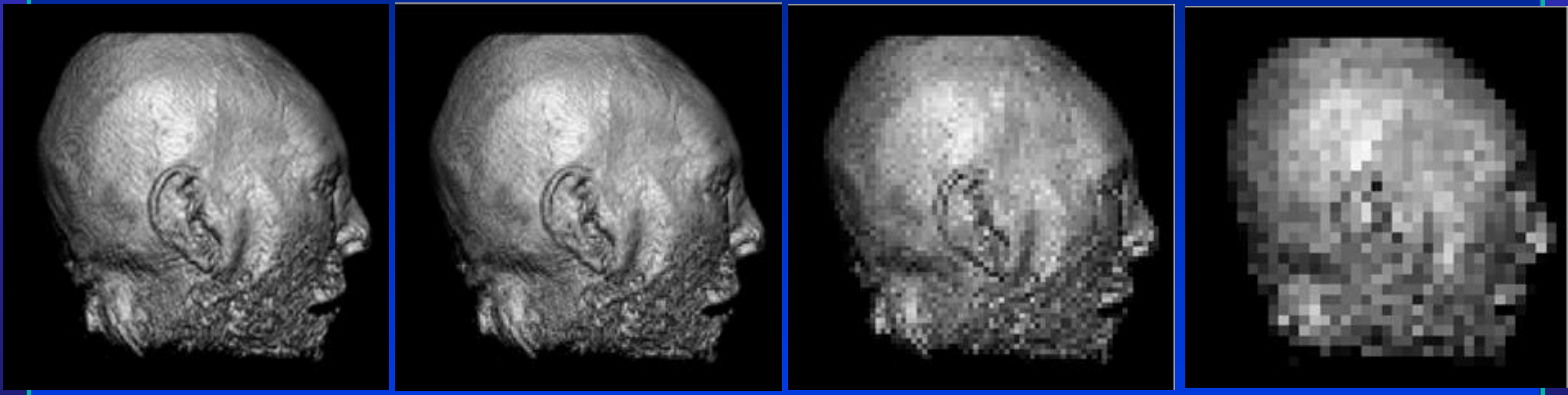


From Human Perception to Digital Images

- Now that we know how the eye perceives light intensity and color, let's take a "look" at digital images
- We'll see what is required in image generation and display in order for the eye to perceive intensities and colors correctly/effectively
- Based on what we learn, we will derive some general principles of color-enabled visualization that will help you in your own visualization-related efforts
  - when is color helpful for understanding
  - when it is necessary for understanding
  - when it distracts from or hampers understanding

Pixels and Images

- Image = 2D matrix or grid of pixels
- Pixel = a dot of light (possibly colored)
- Image resolution = number of pixels along each matrix



res = 300^2 pixels **res = 150^2 pixels** **res = 75^2 pixels** **res =37^2 pixels**

- What is the resolution of a television?
- Why doesn't it look blocky?

Perception of Light Intensity

- The perceived intensity difference of two intensities is a function of the ratio of the two intensities
- Consider a 50-100-150 Watt light bulb: a 50→100 transition is more noticeable than 100→150
- Suppose our monitor has (assume a monochrome B/W monitor) n+1 intensity pixel graylevels, a minimum intensity I_0, and a maximum intensity $I_n = 1.0$
- We would like steps in brightness at all pixel levels *j*: $\frac{I_{j+1}}{I_j} = r = const$
- Thus,

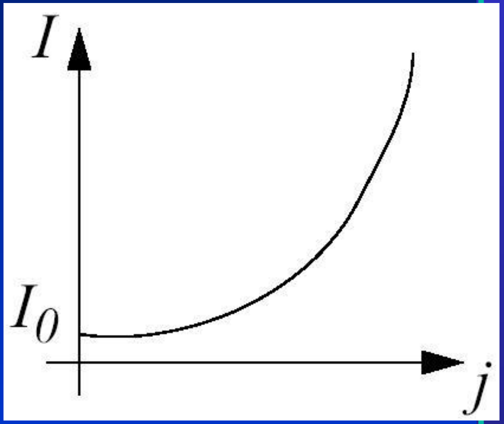
$$I_1 = rI_0 \quad I_2 = rI_1 \quad \cdots \quad (I_j = r^j I_0) \quad \cdots \quad I_N = r^n I_0 = 1.0$$

Perception of Light Intensity

$$I_1 = rI_0 \quad I_2 = rI_1 \quad \cdots \quad (I_j = r^j I_0) \quad \cdots \quad I_N = r^n I_0 = 1.0$$

We can calculate

$$r = \left(\frac{1.0}{I_0}\right)^{\frac{1}{n}}$$



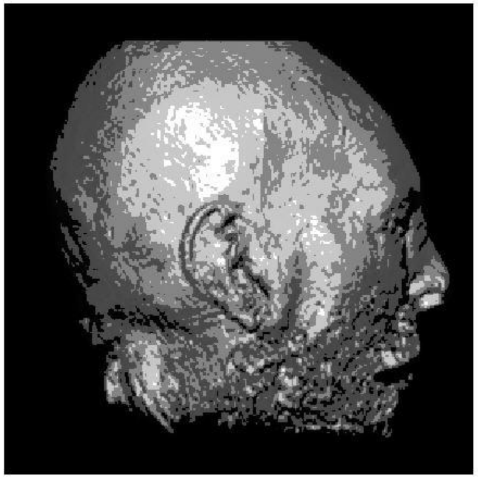
- Experiments reveal that a digital reproduction of an image appears continuous when $r \leq 1.01$
- Thus, the minimum number of intensity levels is:

$$n_{\min} = \log_{1.01}\left(\frac{1.0}{I_0}\right)$$

How Many Intensity Levels Do We Need?

- Given this n_{min}, the eye will just be able to distinguish between I_j and I_{j+1}
- Greater n_{min} are unnecessary since the eye will not be able to distinguish two consecutive gray levels
- Display monitors need 400-530 levels (but usually have only 256)
- Photographic slides have a lower I_0 and thus need more levels (about 700)
- Newspapers have large I_0 and thus can get away with fewer levels (about 230).
- Why do they have large I_0?

8 bit-planes = 256 levels



3 bit-planes = 8 levels

Gamma Correction

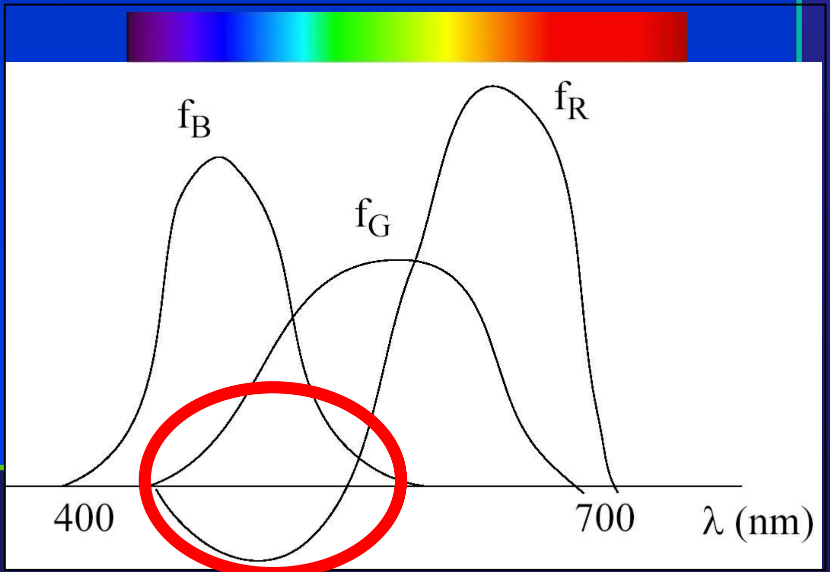
- We'll see now how all this applies to computer displays
- The monitor brightness *I* is related to the number of electrons, *N*, that strike the phosphors: $I = kN^{\gamma}$
- The number of *photons* is non-linearly related to the voltage *V* at the electron gun: $I = KV^{\gamma}$
- We need to correct for this non-linearity: $V = \left(\frac{I}{K}\right)^{\frac{1}{\gamma}}$
- This is called *gamma correction* (for standard monitors, $\gamma = 2.2 - 2.5$, for Trinitron CRTs $\gamma = 1.4 - 1.8$)
- Let's look at a practical example to illustrate how this process actually works

Gamma Correction

- Suppose our pixels have an intensity range of $j = [0, 255]$
- We would like to know the voltage V that we have to apply at the electron gun
  1. Calculate the intensity that yields uniform perception: $I = r^j I_0$
  2. From that, calculate the voltage with proper gamma correction: $V = \left(\frac{I}{K}\right)^{\frac{1}{\gamma}}$
- Most monitors have gamma correction hardware that performs these steps automatically, but often allow changes to be made manually via software
- If done automatically, the hardware stores the pixel values j in the framebuffer and then translates them into the proper voltage

Color Primaries

- Now let's expand our discussion of image generation to color
- As we saw earlier, it is believed that the eye has three kinds of color receptors, one each for red, green and blue
- This is known formally as the *tristimulus theory*
- This means that we can generate nearly any color experience by mixing these three primary colors together on the retina (cones)
- The figure at the right shows the R, G, and B primaries needed to generate a color:
- Notice anything a bit odd about these graphs?



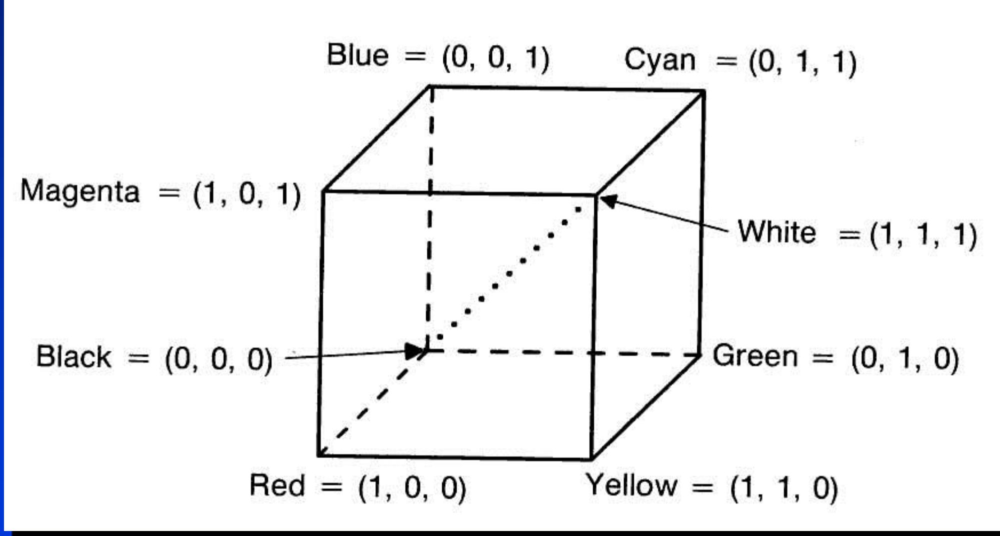
Color Primaries

- The negative part of the graph indicates that the brain does some sort of negative mixing of colors
- This cannot be done with a display monitor that has R, G and B electron guns (note: each gun still fires electrons, not colored light)
- Around 500 nm there are certain colors that cannot be displayed on a computer monitor because a negative amount of red would be required
- Let’s look at how color CRTs work to generate color images

Representation of Color – RGB Color Space

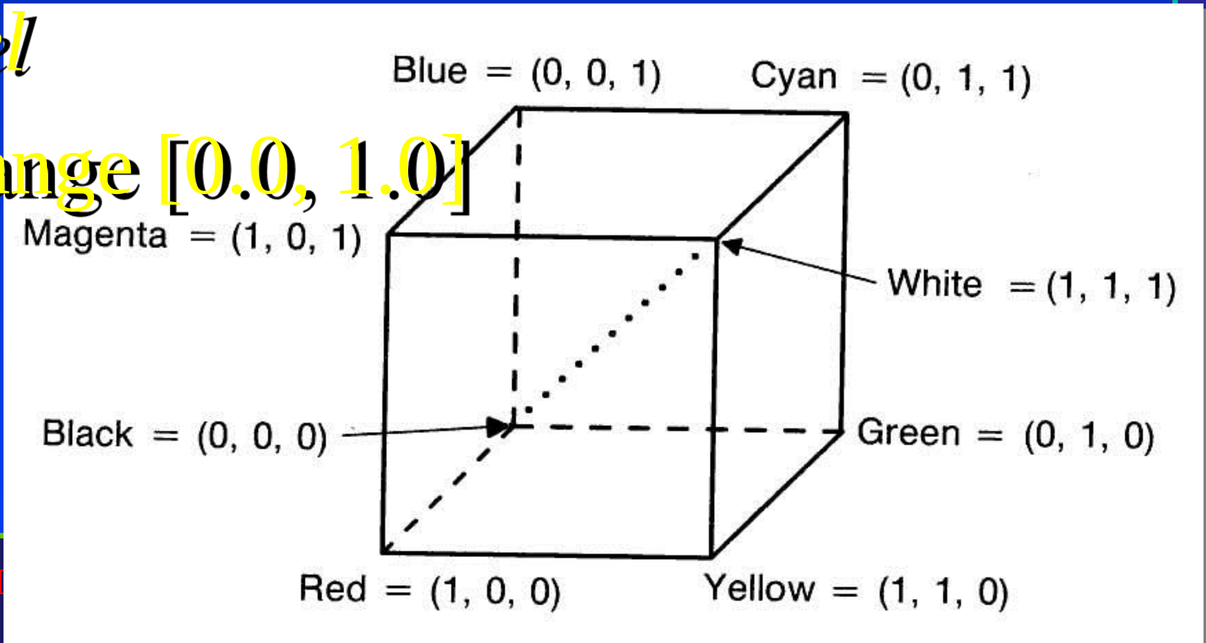
- So how are colors represented in a computer?
- One convenient choice is the RGB *color space*
- R, G, and B in the range [0.0, 1.0]
- black = (0,0,0), white = (1,1,1)
- Complementary colors lie opposite each other on the cube
- Although simple and very fast to use in hardware, the RGB color model is unintuitive when you want to change the saturation (how much gray) and brightness (how much white) is present in the image

- The RGB color cube:



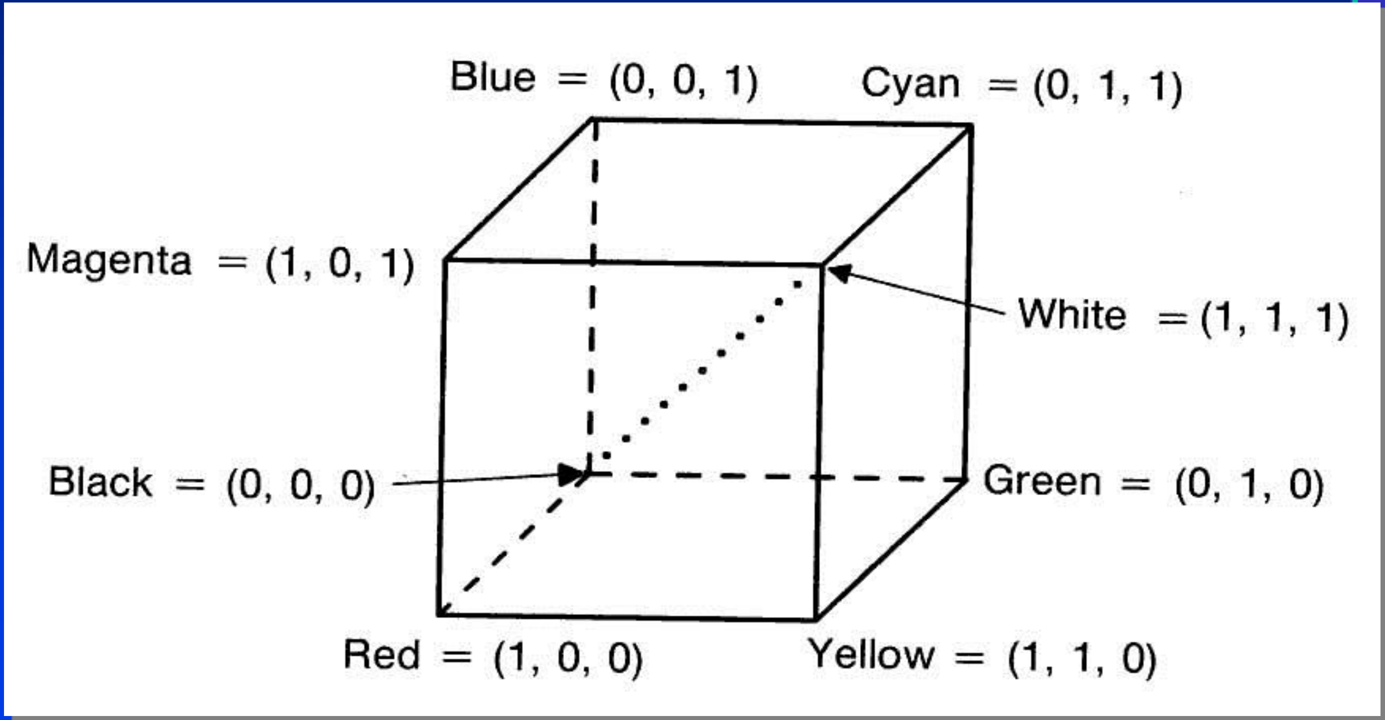
Computer Representation of Color

- Each screen pixel is combination of R, G and B light
- Three *color components* determine perceived color
- *RGB color model*
- R, G, and B in range [0.0, 1.0]



RGB Color Arithmetic

- R + G = ?
- B + G = ?
- R + Y = ?
- R + C = ?
- G + M = ?
- B + Y = ?
- B + W = ?



RGB Color Model

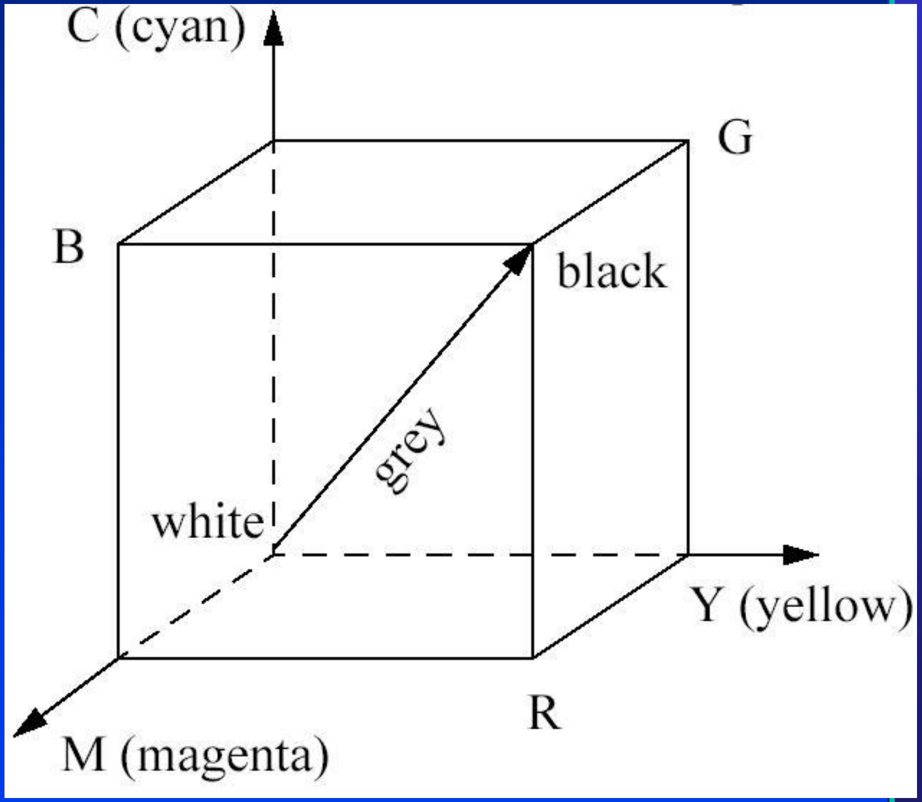
- Good: simple, easy hardware implementation
- Is it intuitive?
- How would you make a washed-out green?
- How would you make a bright blue?

Representation of Color – CMY Color Space

- The CMY color space is the complement of the RGB color space and is used primarily for printing (especially if you add black: CMYK)
- The conversion between RGB and CMY is linear:

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

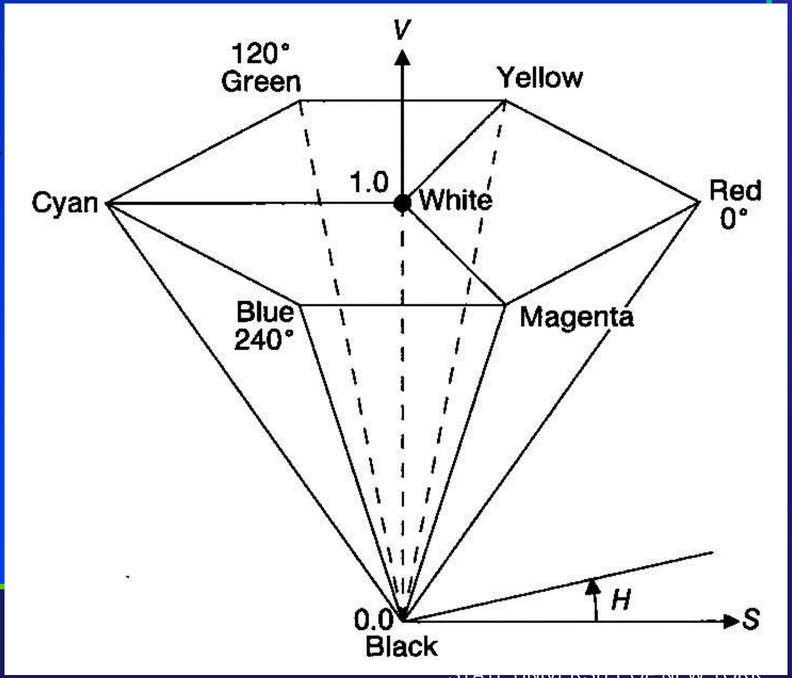
- With CMY, the more C, M and Y you add, the darker the color gets
- With RGB, it's just the opposite
- RGB: *additive mixing with light*
- CMY: *subtractive mixing with pigments*



Representation of Color – HSV Hex Cone

- A more convenient color space for specifying colors is HSV
- Hue, saturation, value (brightness)
- S, V in range [0.0, 1.0]
- H in range [0^o, 360^o]
- Value measures the brightness, or white content
- Saturation measures how vivid or washed out the color is
- Changes in S and V are linear with respect with the HSV color model, but non-linear in RGB
- Easier to specify color in HSV and then translate it into RGB for the hardware

- See Foley pp. 592-593 for algorithms to convert between RGB and HSV

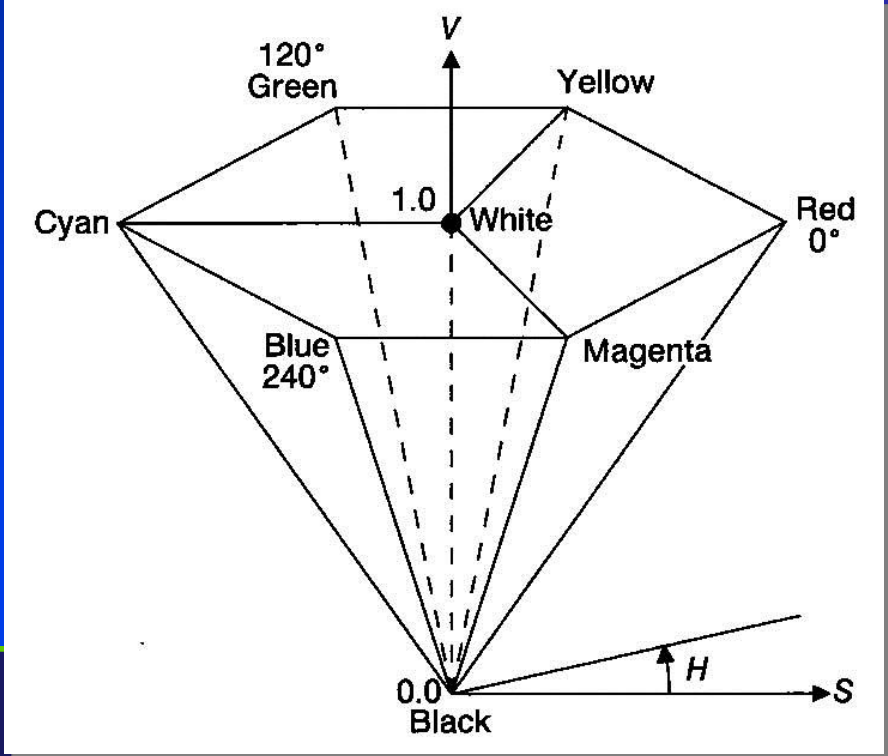


HSV Color Model

- RGB: good for hardware, bad for human use
- Hard to change saturation and brightness
- *HSV color model* – more intuitive
- H = hue
- S = saturation
- V = value (brightness)

HSV Color Model

- *Saturation* measures vividness of color
- Distance from central axis
- *Value* measures brightness
- S, V in range [0.0, 1.0]
- H in range [0^o, 360^o]



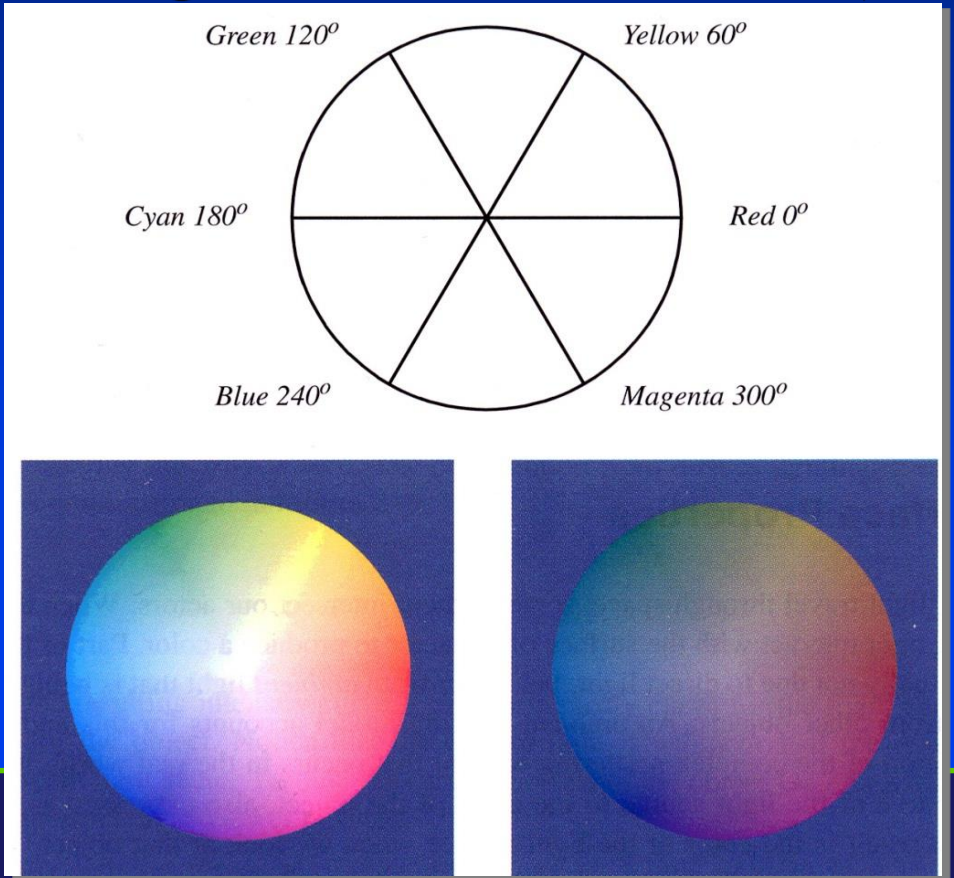
HSV Color Model

- HSV very intuitive
- e.g., to brighten a color, increase V
- e.g., to wash it out (make grayer), decrease S
- How do we change H?
- Change angle
- Sequence of colors around cone:

$$R \rightarrow Y \rightarrow G \rightarrow C \rightarrow B \rightarrow M \rightarrow R$$

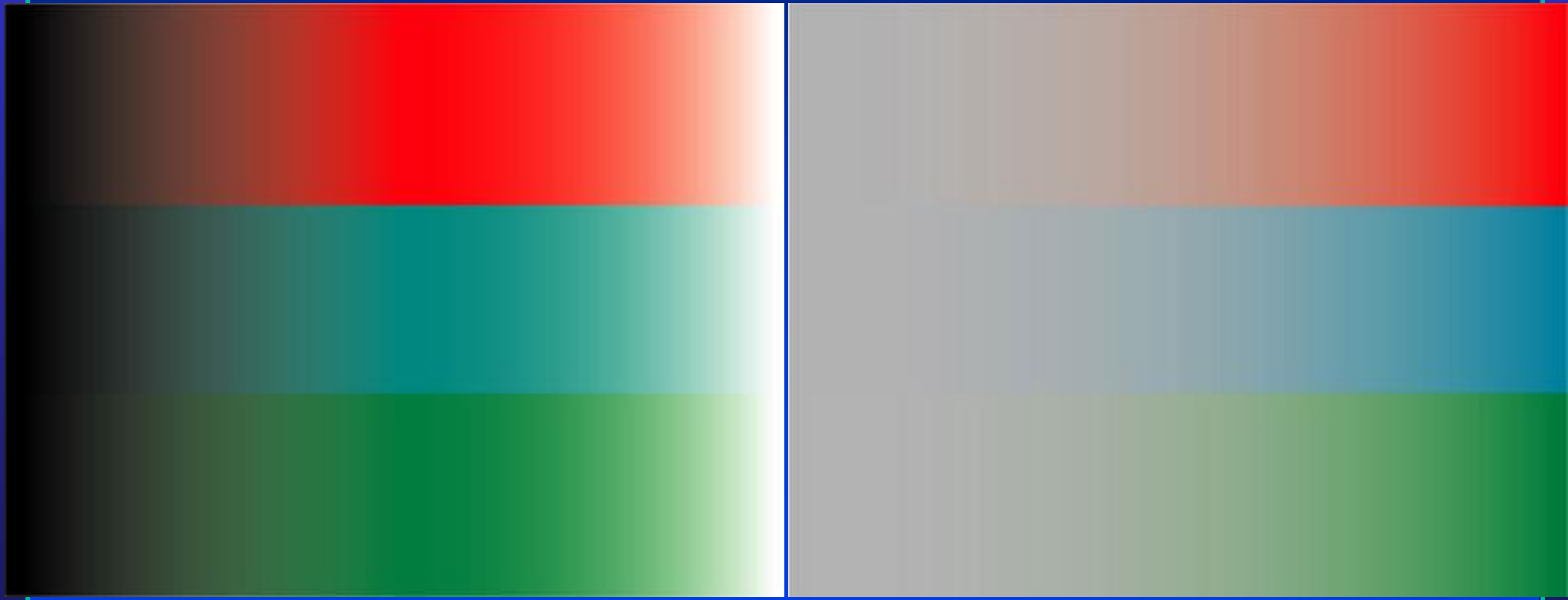
HSV Color Model

- The *hue* is the color
- Specified as angle around the *HSV (hex) cone*



HSV Examples

- Which component of HSV are we increasing in the left image? Right image?



HSV Examples

- Lightness corresponds to how much light appears to be reflected from a surface
- Basically, it measures how much white or black is present
- Saturation is the degree of color intensity associated with a color's perceptual difference from a white, black or gray of equal lightness

RGB vs. HSV

- Can use HSV in software, but convert to RGB for display
- Easy to convert between RGB and HSV
- Code on Web

Color	RGB	HSV
Black		
White		
Red		
Green		
Blue		
Yellow		
Cyan		
Magenta		
Sky Blue		

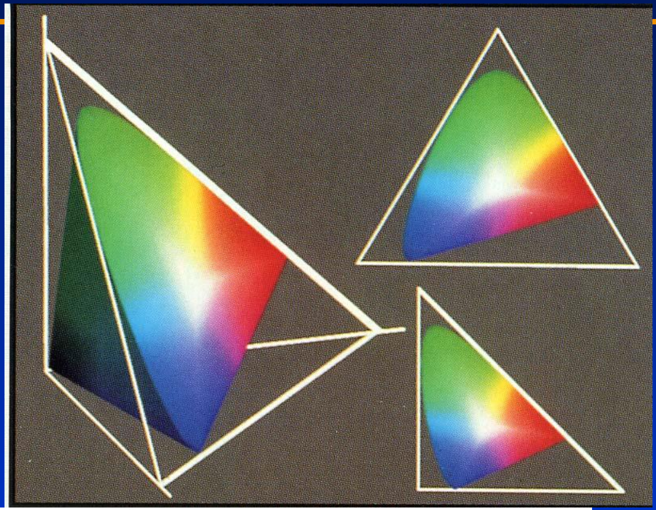
RGB vs HSV

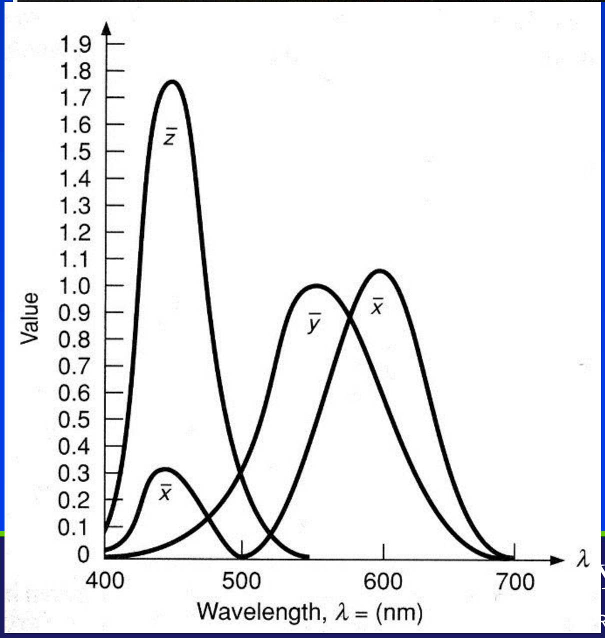
- Can use HSV in software, but convert to RGB for display
- Easy to convert between RGB and HSV
- Code on Web

Color	RGB	HSV
Black	0,0,0	*°,*,0
White	1,1,1	*°,0,1
Red	1,0,0	0°,1,1
Green	0,1,0	120°,1,1
Blue	0,0,1	240°,1,1
Yellow	1,1,0	60°,1,1
Cyan	0,1,1	180°,1,1
Magenta	1,0,1	300°,1,1
Sky Blue	.5, .5, 1	240°,.5,1

The CIE Chromaticity Diagram

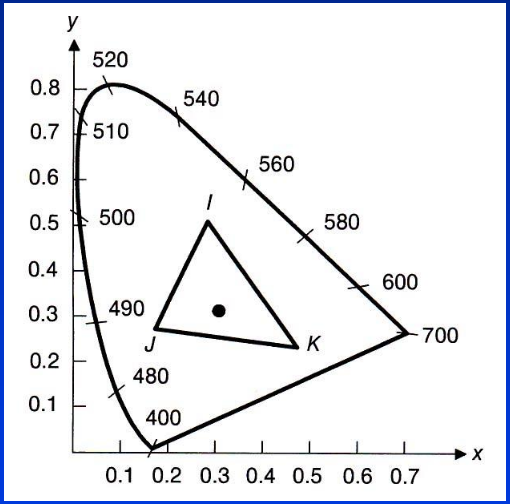
- In 1931, an international commission met to develop a color model that could represent all visible colors *without negative amounts* of red, green or blue
- The primaries they derived are called X, Y and Z
- They are expressed using a set of *color matching formulas*
- The color space has a deformed cone-like shape
- The CIE chromaticity diagram is best used as a tool for studying other color spaces, like RGB and CMY



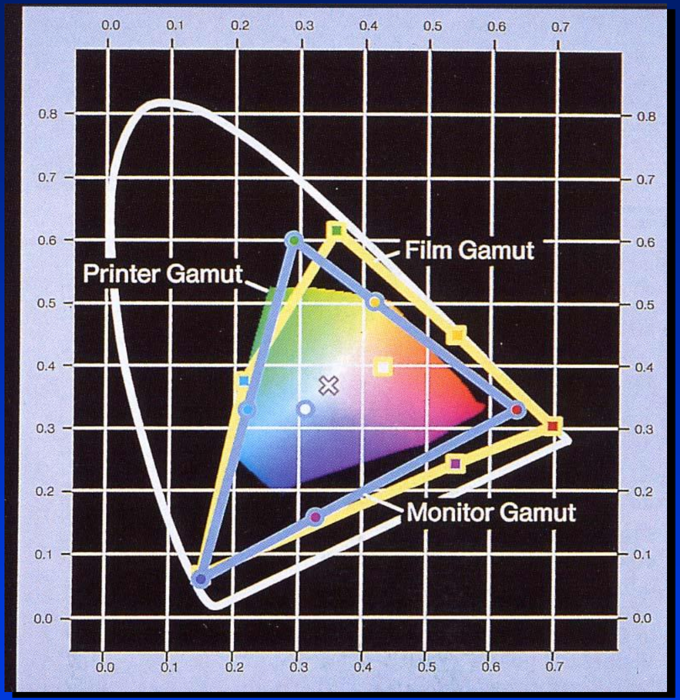


Uses of the CIE Chromaticity Diagram

- The CIE can tell us a lot about *color gamuts*, the range of colors that can be expressed with a particular color model



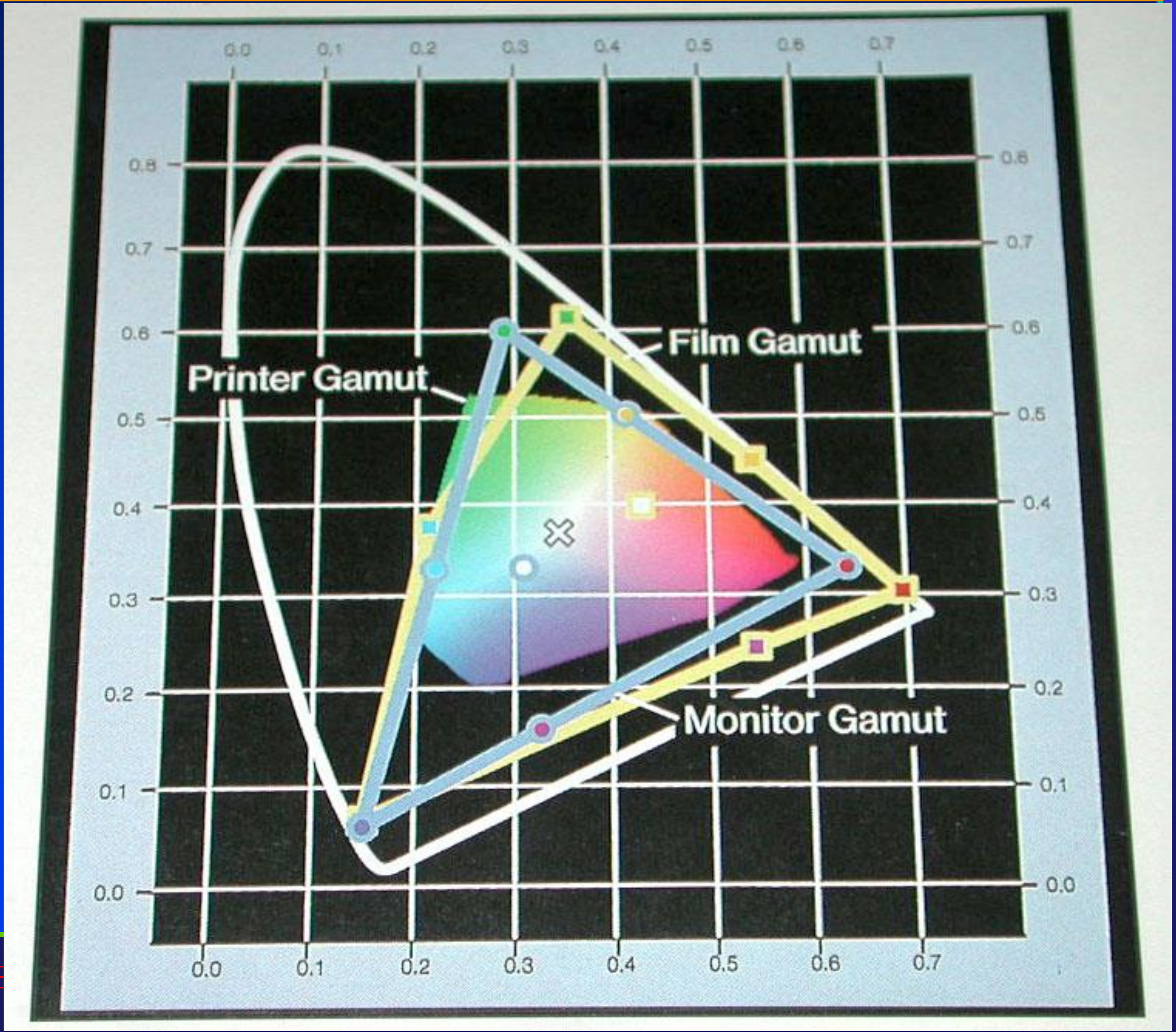
- This figure shows the color gamut for a hypothetical IJK color space
- Note that no single triangle can encompass the entire CIE diagram



- In other words, color monitors and printed materials cannot display the entire spectrum of possible colors

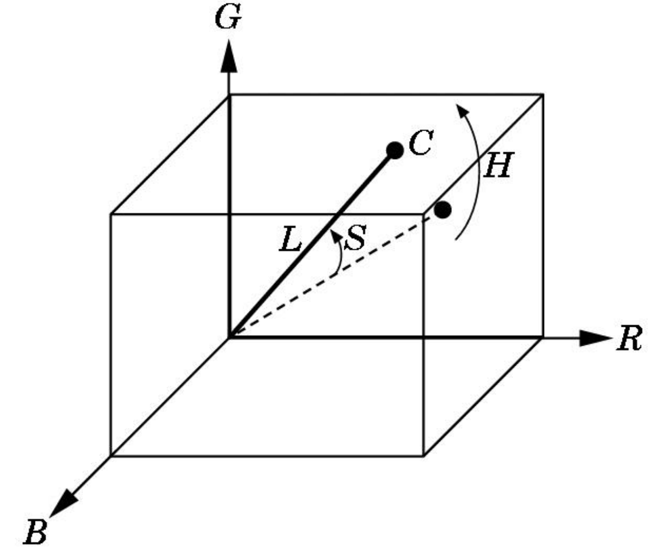
Color Systems

- RGB
- YIQ
- CMYK
- HSV, HLS
- Chromaticity
- Color gamut

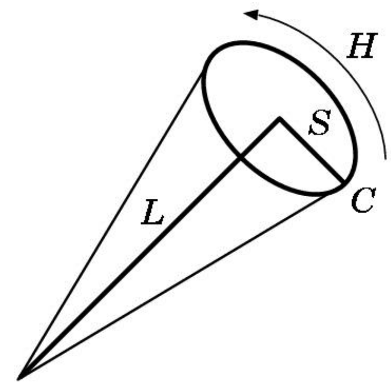


HLS

- Hue: "direction" of color: red, green, purple, etc.
- Lightness: how bright
- Saturation: intensity. E.g. red vs. pink



(a)



(b)

Display Considerations

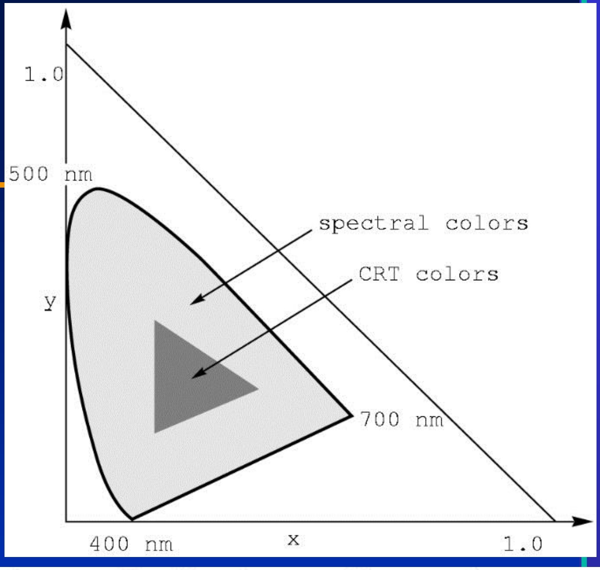
- Color systems
- Color quantization
- Gamma correction
- Dithering and halftoning

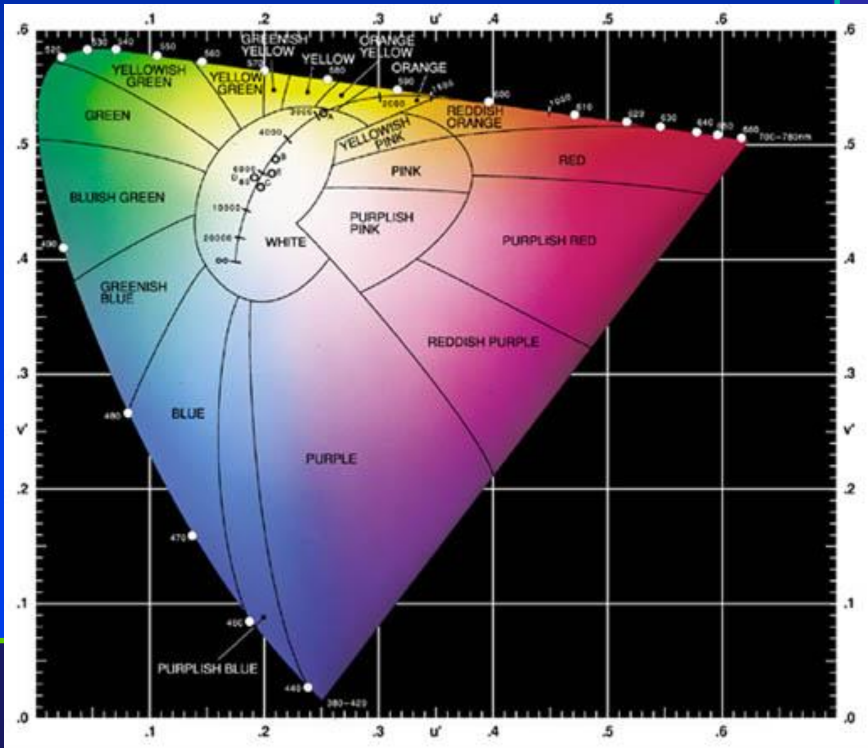
Display Considerations

- Color systems
- Color quantization
- Gamma correction
- Dithering and halftoning

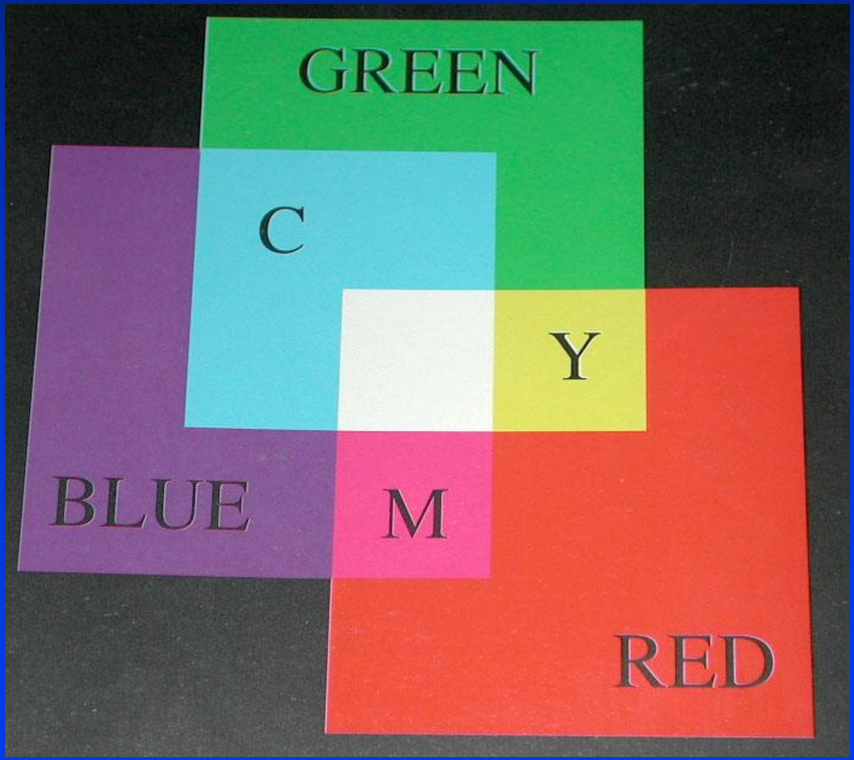
Chromaticity

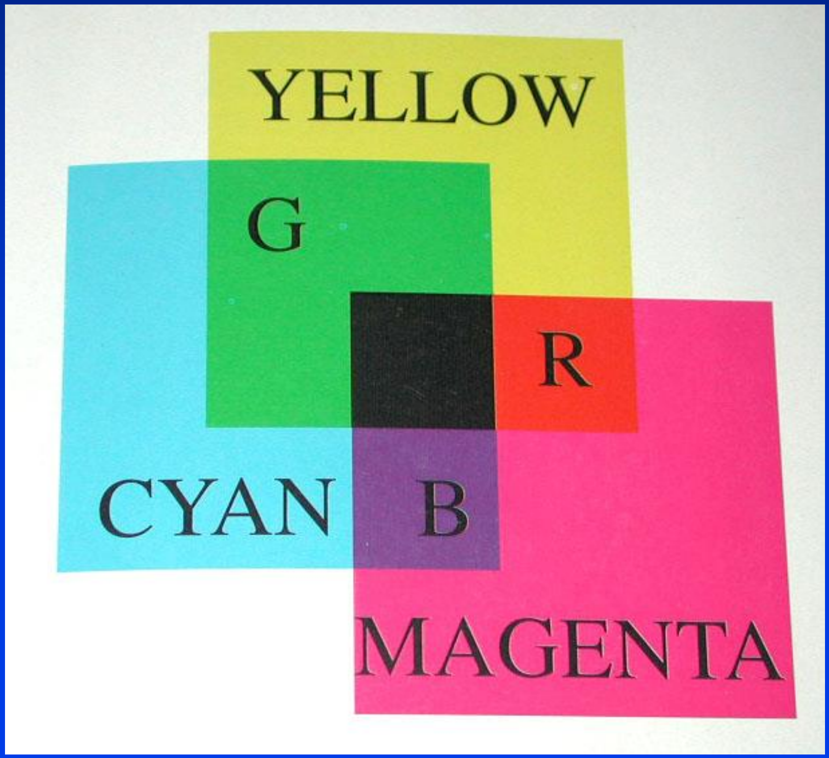
- Tristimulus values: R, G, B values that we know of
- Color researchers often prefer chromaticity coordinates:
  - $t1 = T1 / (T1 + T2 + T3)$
  - $t2 = T2 / (T1 + T2 + T3)$
  - $t3 = T3 / (T1 + T2 + T3)$
- Thus, $t1+t2+t3 = 1.0$
- Use t1 and t2; t3 can be computed as 1-t1-t2
- Chromaticity diagram uses this approach for theoretical XYZ color system, where Y is luminance





Additive and Subtractive Color

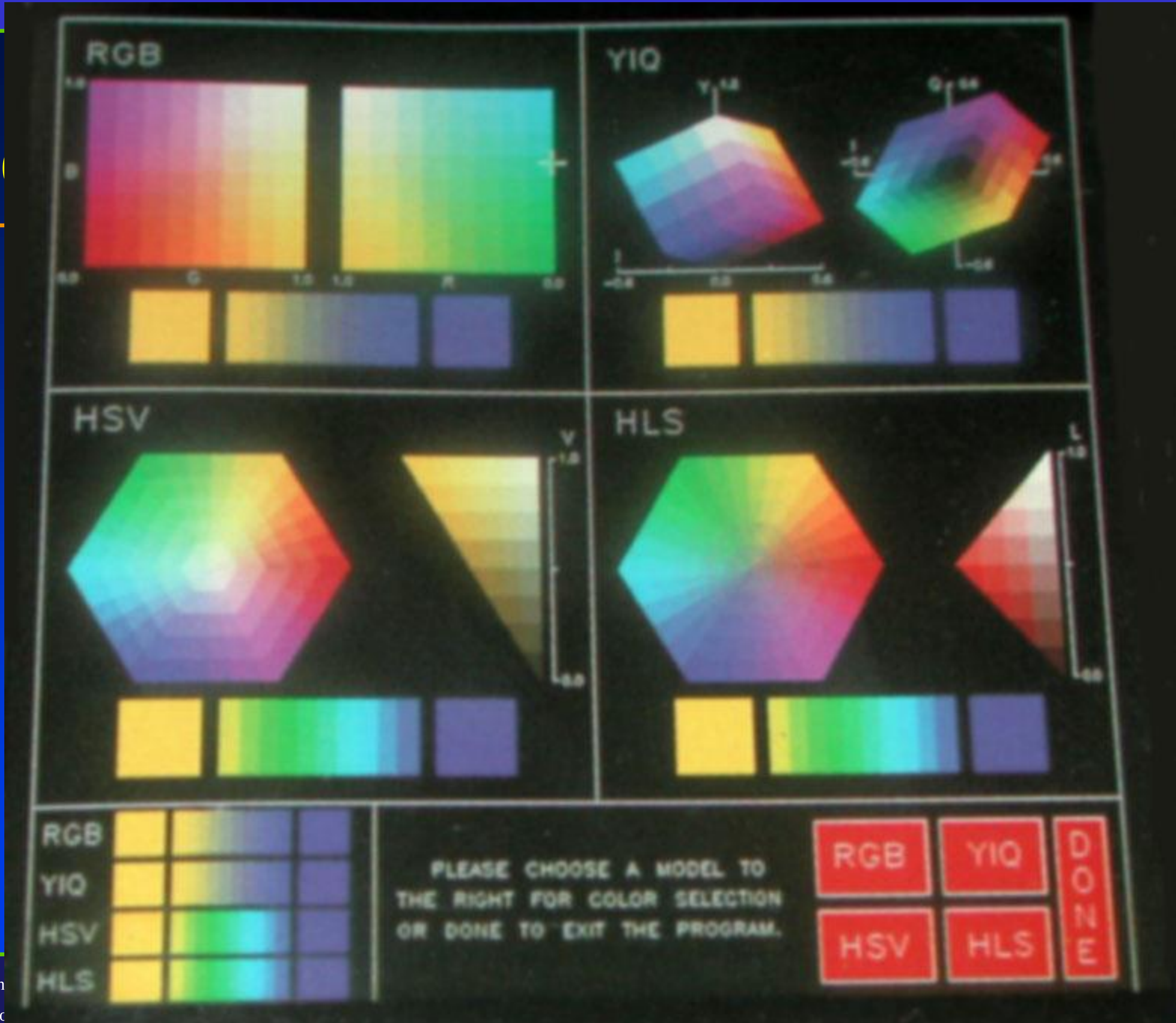




Human Eye Color Perception

- Color resolution - Human eye differentiates about 300 hues and 100-150 luminance variations. What does that mean?
- If red, green and blue cones are 60%, 30% and 10%, which colors can we perceive best?
- Best resolution is for green and red, less resolution for blue
- What does that mean?
- What does this mean for visualization?

C

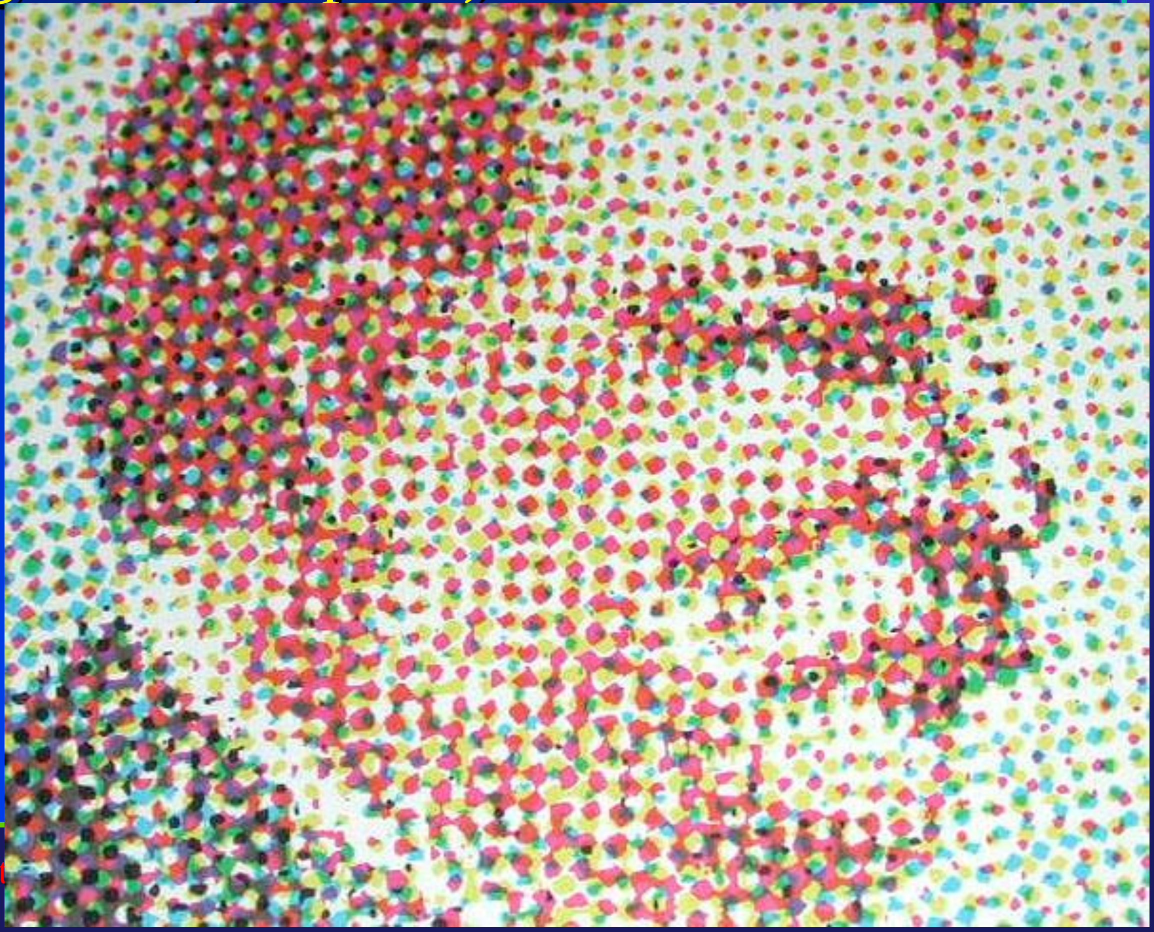

RGB
YIQ
HSV
HLS
RGB
YIQ
HSV
HLS
PLEASE CHOOSE A MODEL TO THE RIGHT FOR COLOR SELECTION OR DONE TO EXIT THE PROGRAM.
RGB
YIQ
HSV
HLS
DONE

Color Quantization

- Color quantization: modifying a full-color image to render with a 256-color palette
- For a fixed palette (e.g., web-safe colors), can use closest available color, possibly with error-diffusion dither
- Algorithm for selecting an adaptive palette?
  - E.g. Heckbert Median-Cut algorithm
    - Make a 3D color histogram
    - Recursively cut the color cube in half at a median
    - Use average color from each resulting box

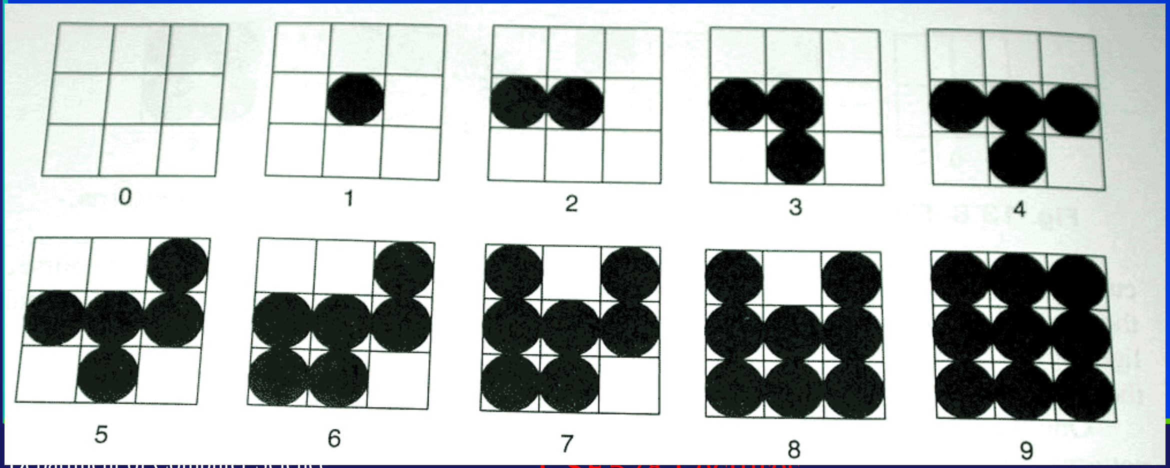
Halftoning

- How do you render a colored image when colors can only be **on** or **off** (e.g. inks, for print)?
- *Halftoning:* dots of varying sizes
- But what if only fixed-sized pixels are available?



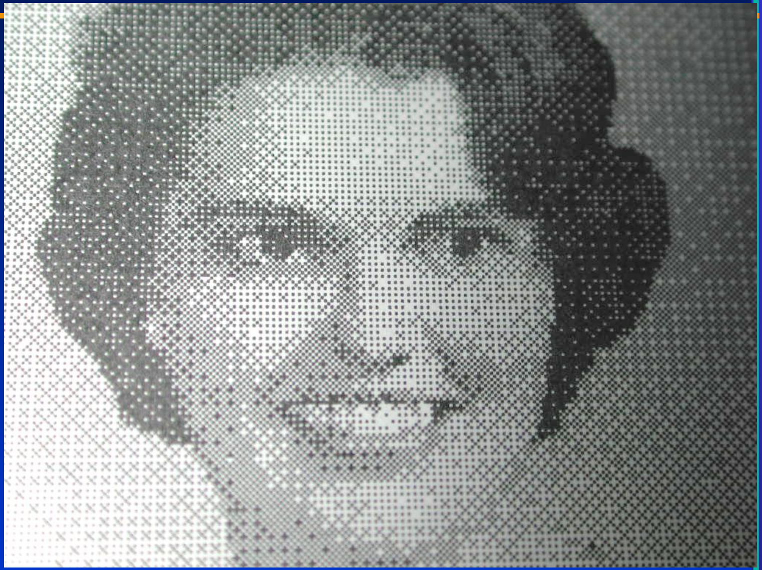
Dithering

- *Dithering* (patterns of b/w or colored dots) used for computer screens
- OpenGL can dither
- But, patterns can be visible and bothersome
  A better approach?



Floyd-Steinberg Error Diffusion Dither

- Spread out "error term"
  - 7/16 right
  - 3/16 below left
  - 5/16 below
  - 1/16 below right
- Note that you can also do this for color images (dither a color image onto a fixed 256-color palette)



Human Eye Color Response

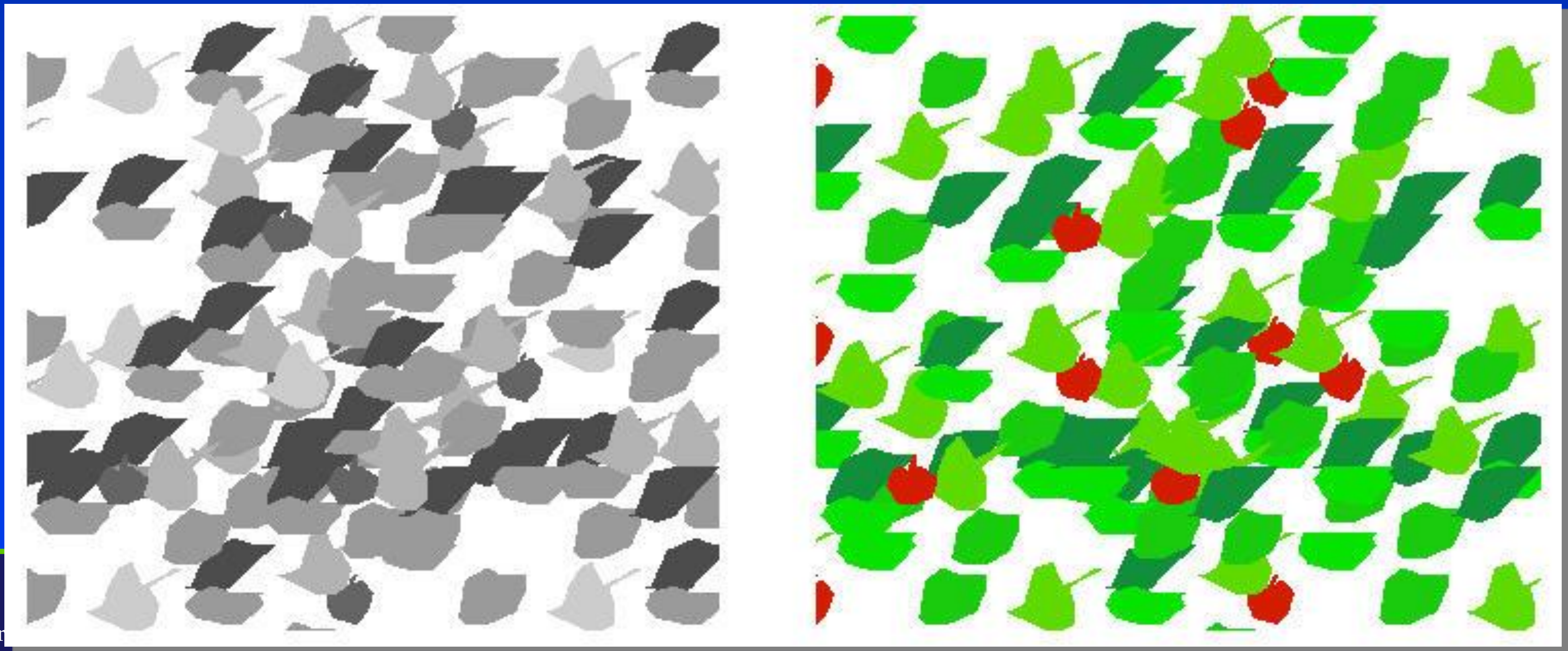
- The time to response to a signal varies according to the color used
- Human eye responds to certain colors faster than others
- Color ranking (from best to worst):
  yellow > white > red > green > blue
- What colors should be used to highlight important features? - Important features should be visualized in light colors, such as yellow and white
- Background information is best visualized in dark colors, such as green and blue
- Let's test your color response

Human Eye Color Perception

- Color channel properties:
  - luminance channel: detail, form, shading, stereo, motion
  - color: surfaces of things, labels, categories (about 10)
  - red, green, blue, yellow are special
- Chromatic channels have low resolution
  - luminance contrast needed to see detail (3:1 recommended, 10:1 for small text)

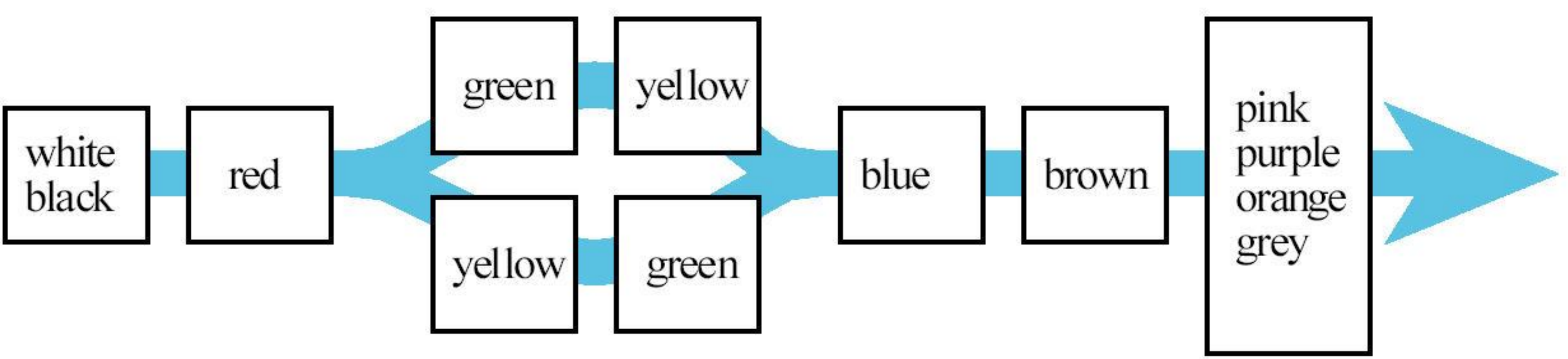
Information Coding with Color

- Color good for *classification* – separation of data into classes
- Color helps us determine type and character of data
- In practice, only about six categories can be distinguished using color alone
- Keep this in mind when developing your own visualization of datasets



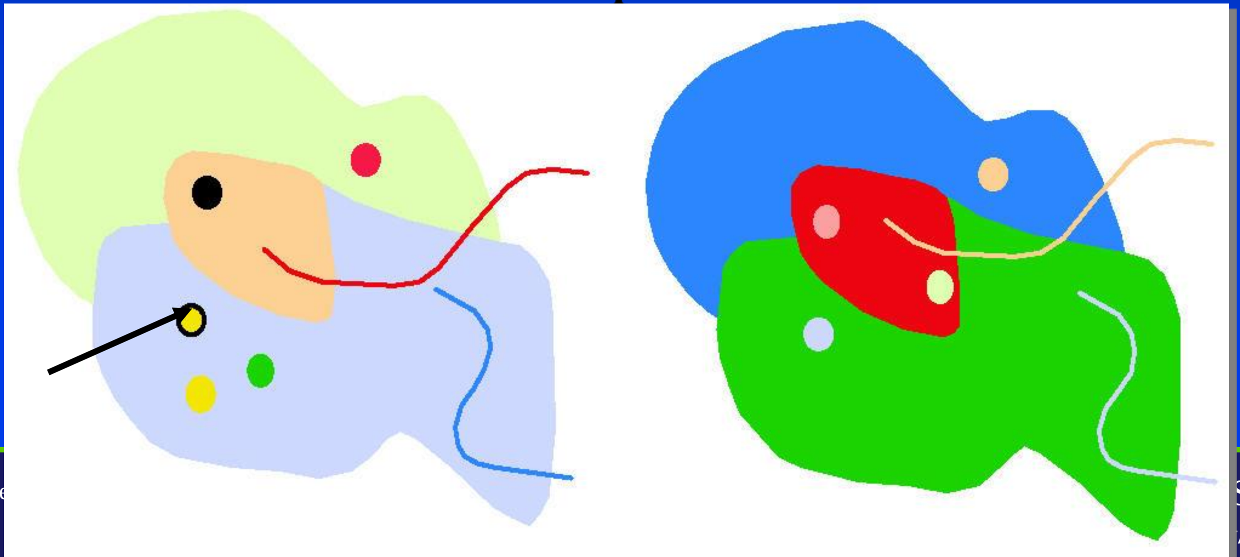
Information Coding with Color

- A suggested order for adding color to visualization



Information Coding with Color – Helpful Tips

- Color coding
  - large areas: low saturation
  - small areas: high saturation
  - maintain luminance contrast
  - break iso-luminances with borders (?)
- Examples of these ideas in practice:

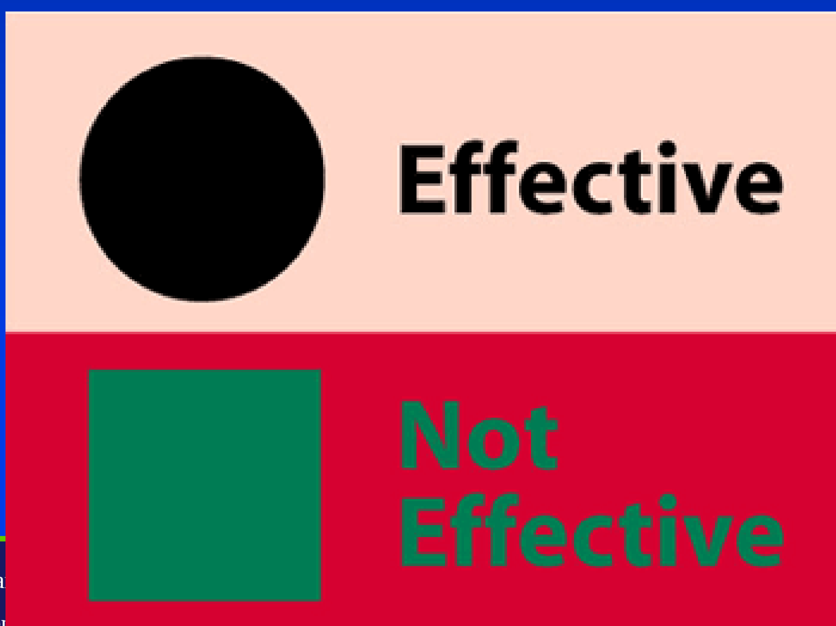


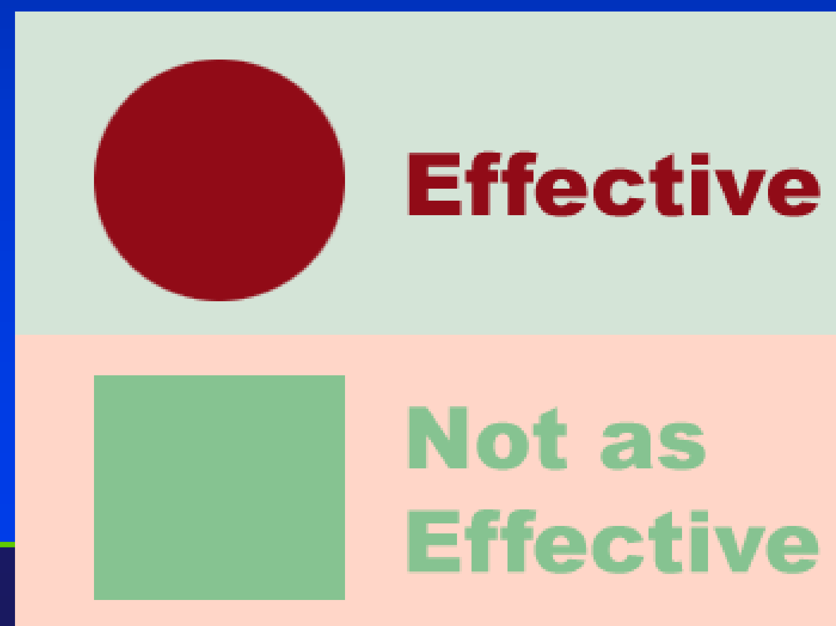
Human Eye Color Perception

- We are sensitive to small color differences
- Good at making side-by-side comparisons
- Not as good at identifying colors in isolation
- Hard: “Is that red, or orange-red, or red-orange, or maroon or….?”
- Easier: “Color A is redder than color B”

Color Response and Perception Summary

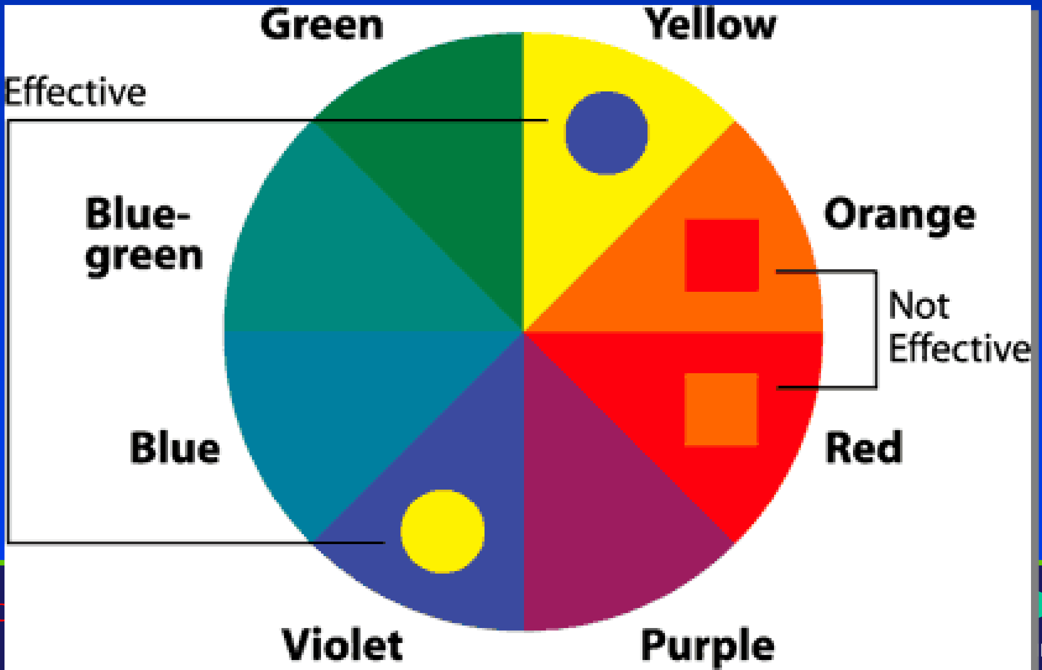
- Use bright colors to highlight important features
- Yellows, oranges, reds will be picked up first by the eye
- Make effective use of light/dark contrast:





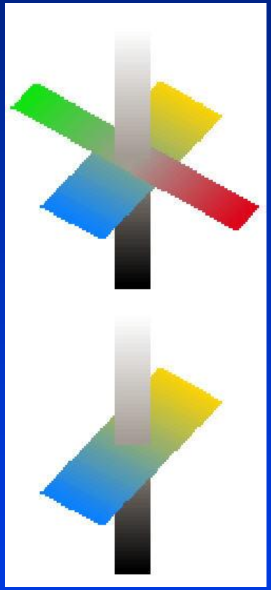
Color Response and Perception Examples

- The human eye is very good at making side-by-side comparisons
- Also make effective use of color contrast to highlight important and interesting characteristics of data



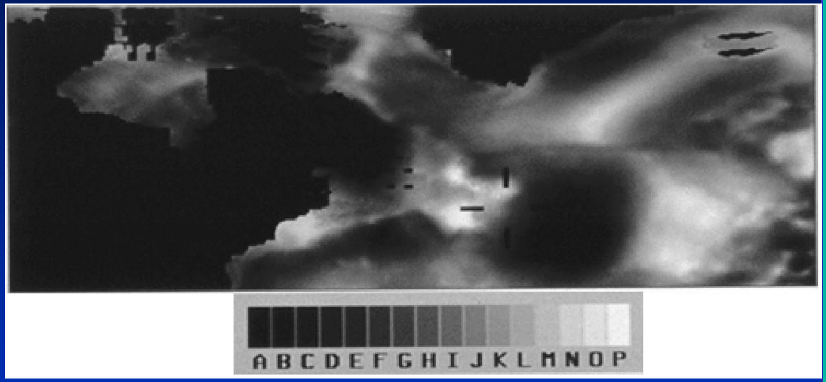
Color Perception

- Color blindness collapses the 3D color space into 2D
- 8% of males are color-blind
- Color resolution
  - color perception is relative
  - we are sensitive to small differences, meaning that we need millions of colors in practice
  - we are best able to distinguish colors when making side-by-side comparisons
  - however, we are not sensitive to absolute values, meaning that we can use only < 10 color values for coding information

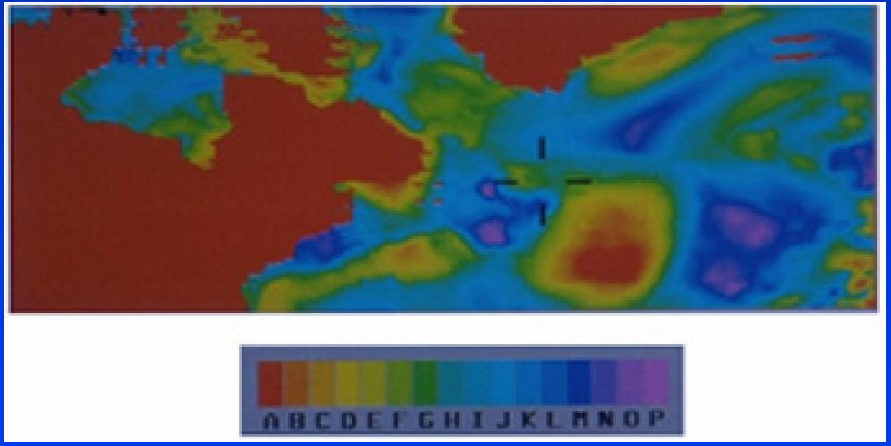


Pseudo-coloring (Colorization)

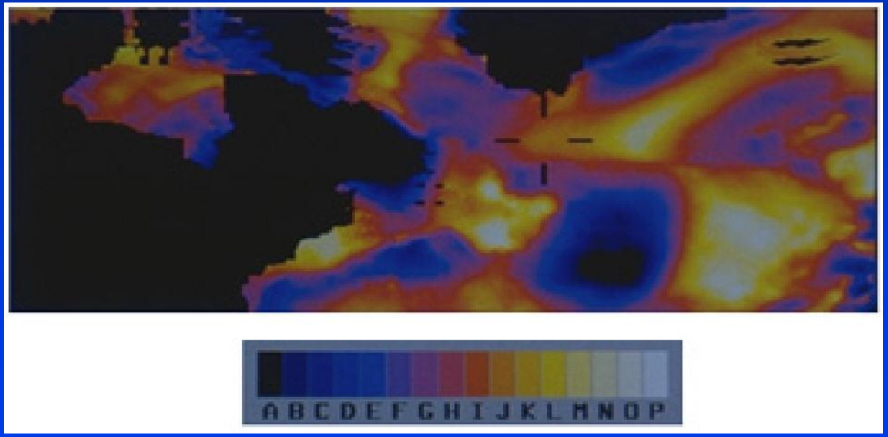
- Assign colors to gray levels by indexing the gray levels into a color map
- This indexing or translation takes place via a *transfer function*
- This is a simple yet powerful technique used in many types of visualization



original graylevel map



simple spectrum sequence with iso-luminance



more effective: spiral sequence through color space; luminance increases with hue

Color Perception - Summary

- Use luminance for detail, shape, and form
- Use color for coding - few colors
- Minimize contrast effects
- Strong colors for small areas - contrast in luminance with background
- Subtle colors can be used to segment large areas
- Figure at right shows how to use variation of luminance to indicate the direction of flow