# COLOR THEORY

# TOPICS TO BE COVERED

**The color problem**

**Cone response and Tristimulus Vectors**

**Solution to the Color problem – using CMFs**

**Chromaticity Diagrams**

**Color Spaces and Examples**

**Displays**

**Color Quantization**

# THE COLOR PROBLEM

**How do we perceive an object?**
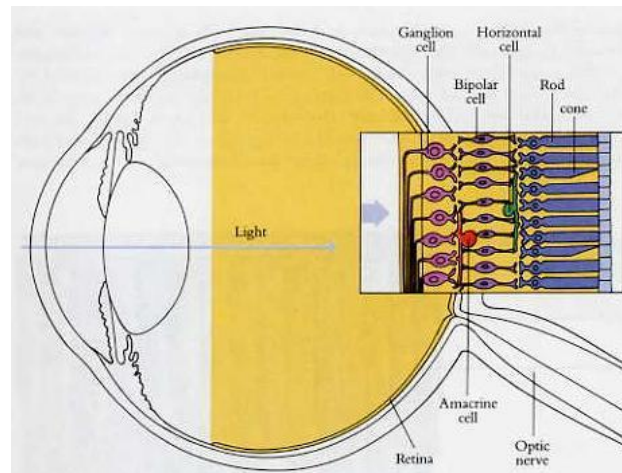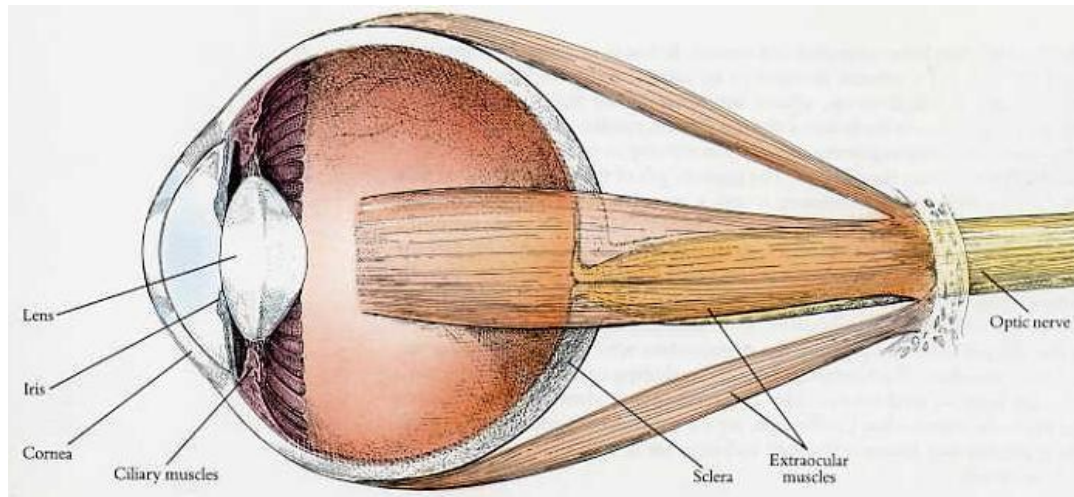
**How do we capture the image of an object?**
- **Digital Still Camera**
- **Camcorders**
- **Film processing/scanning**
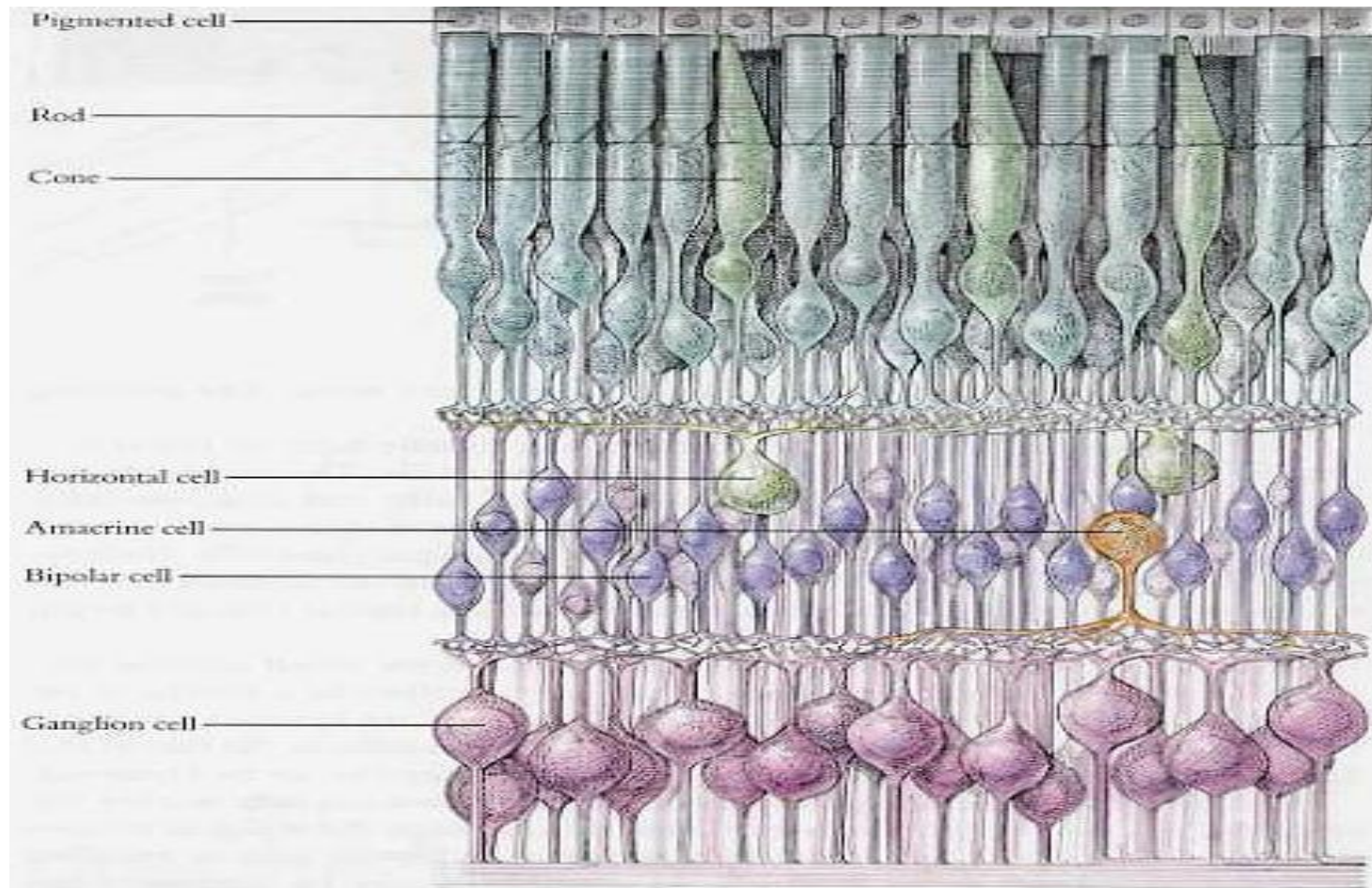
**How do render an object image**
- **Storage/Transfer issues**
- **Rendering on Client**
- **Printing**

**The Color Problem – How do you ensure that the colors in the image of a rendered object look the same as the original colors of that object?**
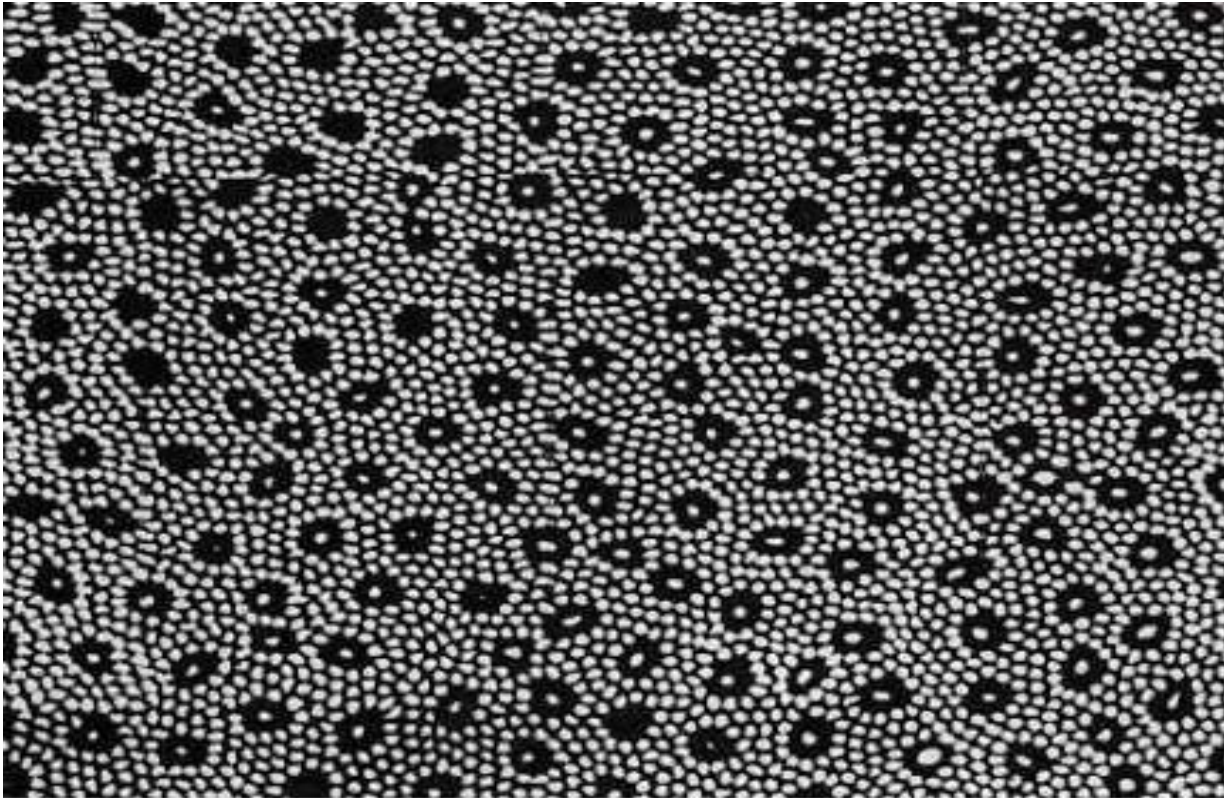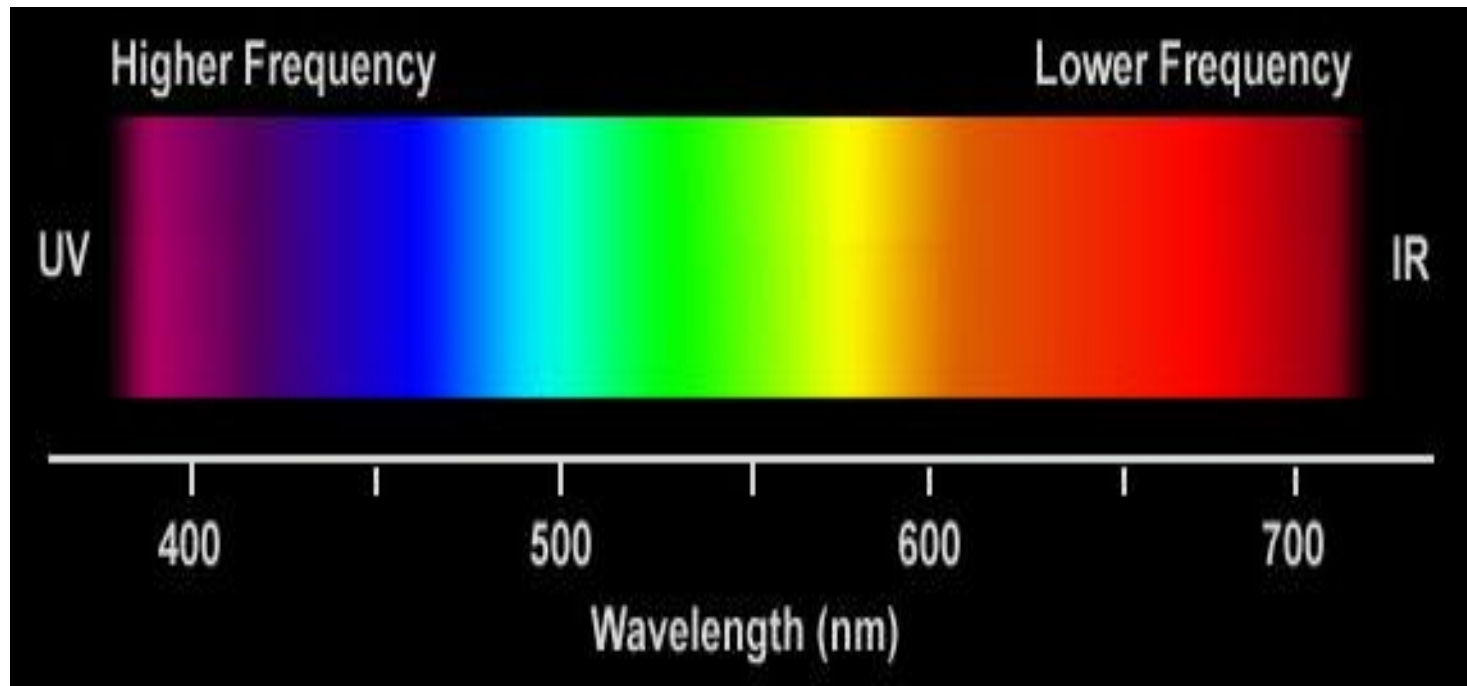
# EYE STRUCTURE

# RETINA



Pigmented cell

Rod

Cone

Horizontal cell

Amacrine cell

Bipolar cell

Ganglion cell

# RETINA

# HOW DO WE PERCEIVE COLORS?

**Spectrum of visible light $f(\lambda)$: $\lambda$ =360-830 nm**
    **($\lambda$ = wavelength)**

**In the *retin*a: 3 types of cones with different spectral absorptance (or sensitivity). The response $c_i$ of a cone of type *i* is modeled as**
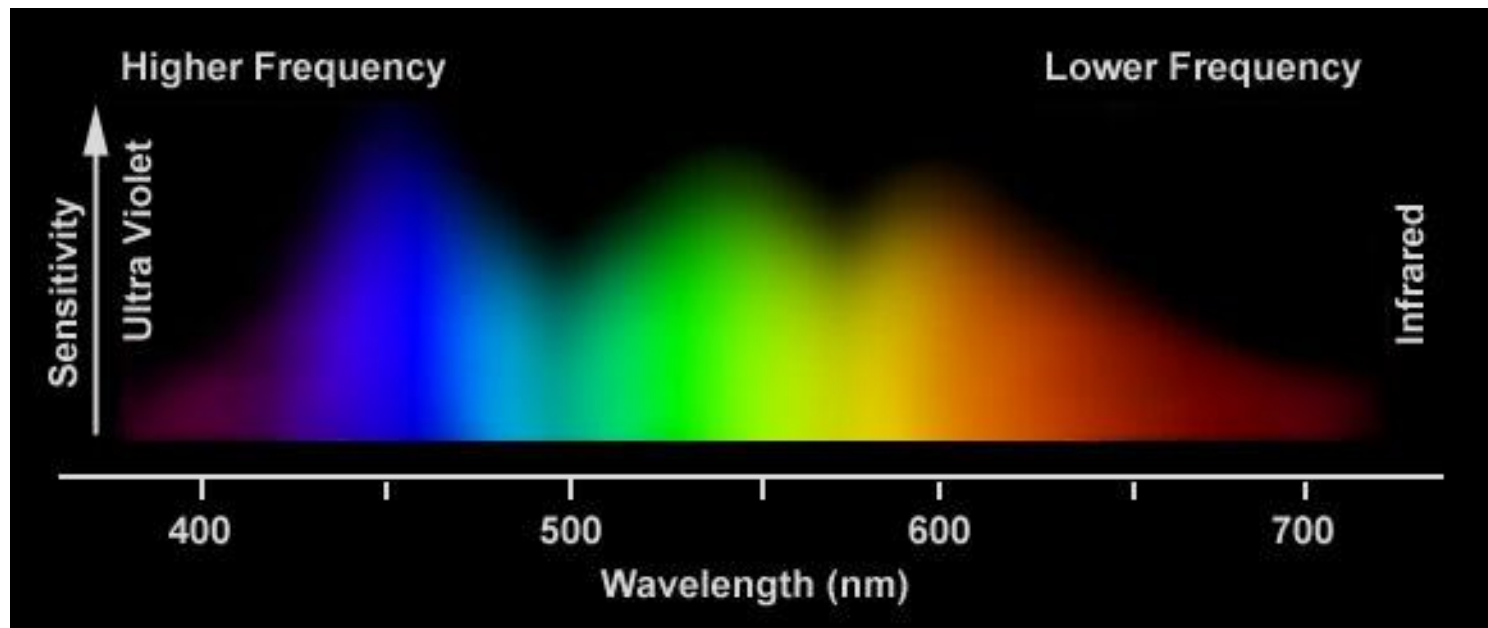
$$c_i = \int_{\lambda_{\min}}^{\lambda_{\max}} s_i(\lambda) f(\lambda) d\lambda$$
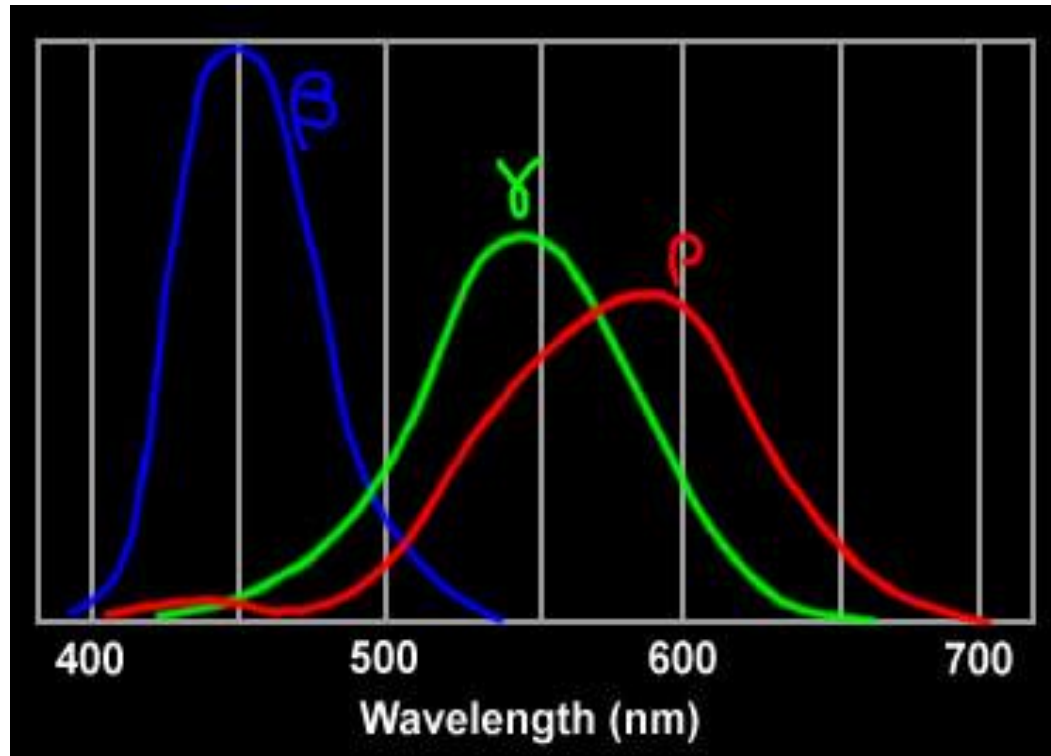
# SPECTRAL DISTRIBUTION

# SPECTRAL SENSITIVITY

**Sensitivity for Entire Spectrum**

# SPECTRAL SENSITIVITY OF CONES

# DISCRETE FUNCTION EVALUATION OF CONE RESPONSE

Instead of evaluating the cone response as a continuous integral function, we will use discrete quantities with summations

- Represent the light spectral stimulus $f(\lambda)$ as a vector $f$ on *N* different wavelength (with a spectral interval of 10 nm)
- Represent each sensitivity $s_i(\lambda)$ as vector $s_i$ as *N* different wavelength

Then the cone response is represented by a vector (called tristimulus vector):

$$C = S^T f$$

# TRISTIMULUS VECTORS

$C = S^T f$ is a 3-vector (called *color*) which represents how we *perceive* a light with spectrum f (which is a N-vector).

If two different spectra $f_1$ and $f_2$ produce the same color $C = S^T f_1 = S^T f_2$, then they *look the same*, that is, they are undistinguishable by the human eye.

Thus, it is *not* important that after processing, transmission and reproduction we recreate the same light spectrum f as the original, but that we reproduce a light spectrum that has the same perceived color $c$.

Thus, 3 values (the entries of C) are sufficient to represent the color of a pixel (instead of *N* values)

# ISSUES WITH GENERATING COLOR

The tristimulus or color equation is $C = S^T f$
- We need to know S
- We need to know f

We don't actually know S; we will show soon that it doesn't matter

To understand f we need to know
- How should we capture a light spectrum? (How do we take a color picture?)
- How do we reproduce a light spectrum? (How do we render a color?)

# HOW DOES A 3-CCD CAMERA WORK?

**Light is refracted onto 3 CCDs (through a prism). In front of each CCD there is a *color filter* with frequency response $m_i(\lambda)$. Thus, a pixel of the i-th CCD gives an output**

$$a_i = \int_{\lambda_{min}}^{\lambda_{max}} m_i(\lambda) f(\lambda) d\lambda$$

**In discrete form: a = M$^T$f where the i-th column of M contains the samples of $m_i(\lambda)$**

**Note that M can only have positive entries!**

# HOW DOES A COLOR PROJECTOR WORK?

Color Projectors work using Additive Synthesis: there are actually 3 light projectors focusing light onto the same spot.

Let the light spectra produced by each be: $p_1$, $p_2$, $p_3$. The intensity of each projector can be controlled at each pixel via a *gain* $a_i$. Thus, the actual light spectrum on the screen is $g = p_1a_1 + p_2a_2 + p_3a_3 = Pa$
    where $P=[\ p_1, p_2, p_3\ ]$ (size: Nx3) and
    $a=[\ a_1, a_2, a_3]^T$ (size:3x1)

Are 3 projectors able to create all the *light spectra* that we want to reproduce? No! But we just need that they create the *colors* we need to reproduce.

# COLOR PROJECTOR (2)

Let { $p_1$, $p_2$, $p_3$ } by three "colorimetrically independent" light sources (called *primaries*).

Then, given any spectrum f, we can always find gains a such that the light g = Pa generated by additive synthesis has the same *color* of f.

*Proof*: We need to find a vector a such that
$$S^Tg = S^TPa = C.$$ Just choose $a=(S^TP)^{-1}C$.

*Problem*: a may have negative components, which makes no physical sense (gains and spectra can only have positive values!)

# THE PROBLEM

**Given a set of primaries P, how can we be sure that for any light spectrum f, the projector reproduces a light spectrum with the same color of f?**



$$a = M^T f$$

$$g = Pa$$
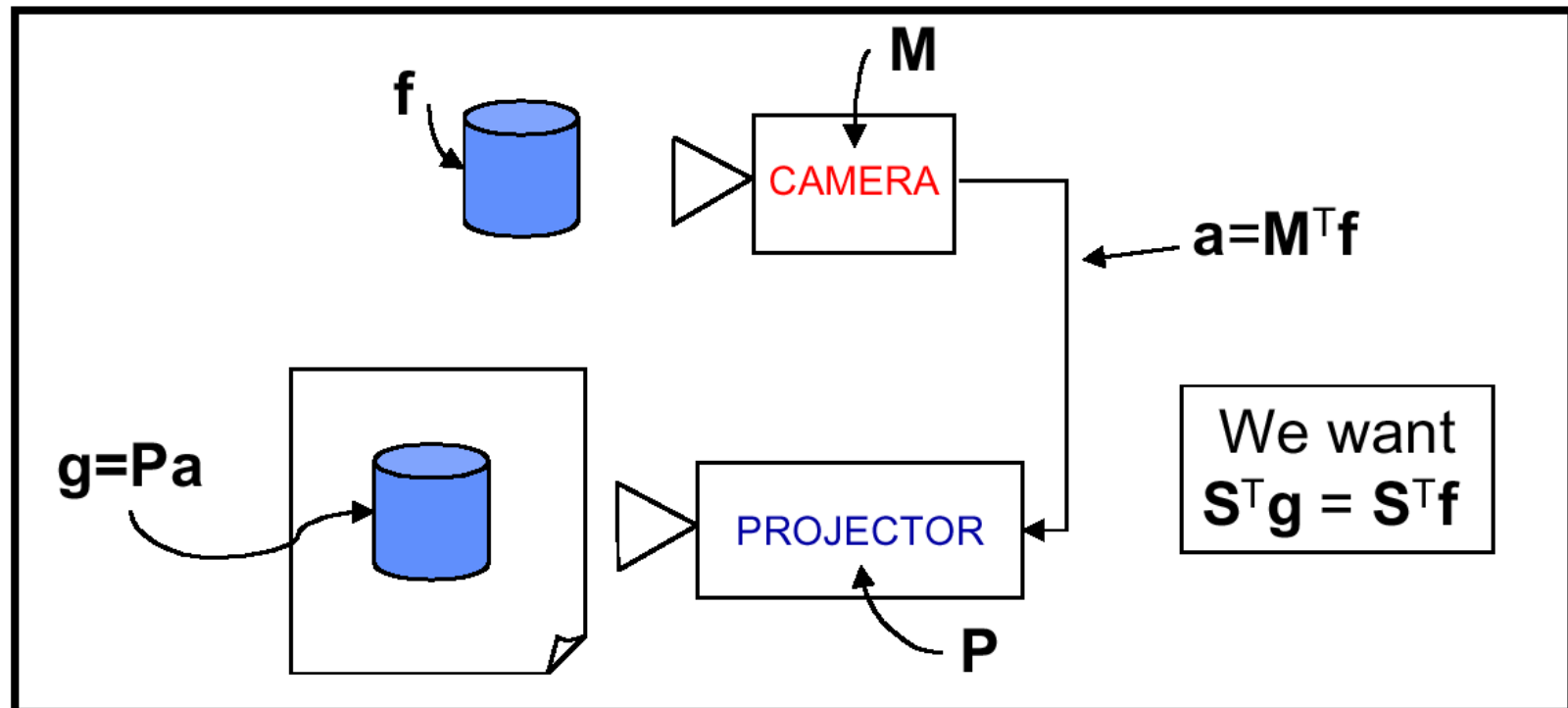
We want
$$S^T g = S^T f$$

# THE PROBLEM (2)

Given a set of primaries P, design the "filters" M of a color camera such that for any input spectrum f, the light g generated by the projector has the same color of f.

In other words: we want that the projectors, using as gains the output a of the camera, create a light $g = Pa$ such that $S^T g = S^T Pa = S^T f = C$.

And we want to do it without knowing S (which is difficult to compute)

# PROBLEM SOLUTION

The solution is based on color matching experiments

The columns { $m_1$, $m_2$, $m_3$ } of M (size: $Nx3$) are called the color matching functions (CMF) associated with the primaries P.

Problem (again): CMF can take on negative values.

# WHAT ARE COLOR MATCHING FUNCTIONS?

Like we said before: the three CMFs { $m_1$, $m_2$, $m_3$ } correspond to the " filters" in a color camera to generate the three color channels a = { $a_1$, $a_2$, $a_3$ }.

Each color signal can be used to control the intensities of the primaries (projectors) in order to generate a spectrum g = Pa that has the same color as f

$$S^T g = S^T Pa$$

Note: Given a set or primaries P, there is just one set of CMFs M associated to it, and vice-versa

Remember, C is the "perceptual" color vector; a is the color vector associated to the chosen primaries

# CHANGING PRIMARIES

What happens when we change primaries? (e.g., we use a different projector, or we use a CRT screen instead of a projector)

Suppose we use primaries Q instead of P. We want to find a new color vector b such that
$$S^TPa = S^TQb$$

The solution is b=Ta with $T=(S^TQ)^{-1}(S^TP)$
    (T is a 3x3 matrix which is independent of a)

Thus T is a linear operator that maps from one color space into a new one.

# CIE CMFS

The International Commission on Illumination (CIE) has standardized two sets of CMFs: *RGB* and *XYZ*

RGB: The CMF's are associated with *monochromatic primaries* at wavelengths of 700, 546 and 435 nm.
(Note that one RGB CMF takes on -ve values.)

XYZ: One of the 3 CMFs, y($\lambda$), is the *luminous efficient function*, which gives the relative sensitivity of the eye to each wavelength. *Y* is called *luminance*. (Note that all *XYZ* CMFs are positive; however, it can be shown that the corresponding primaries are not physically realizable)

Note: *{r($\lambda$),g($\lambda$),b($\lambda$)}* (*{x($\lambda$),y($\lambda$),z($\lambda$)}*) are the CMFs (corresponding to the columns of M); the values *{R,G,B} {X,Y,Z}* are the color channels (corresponding to a)
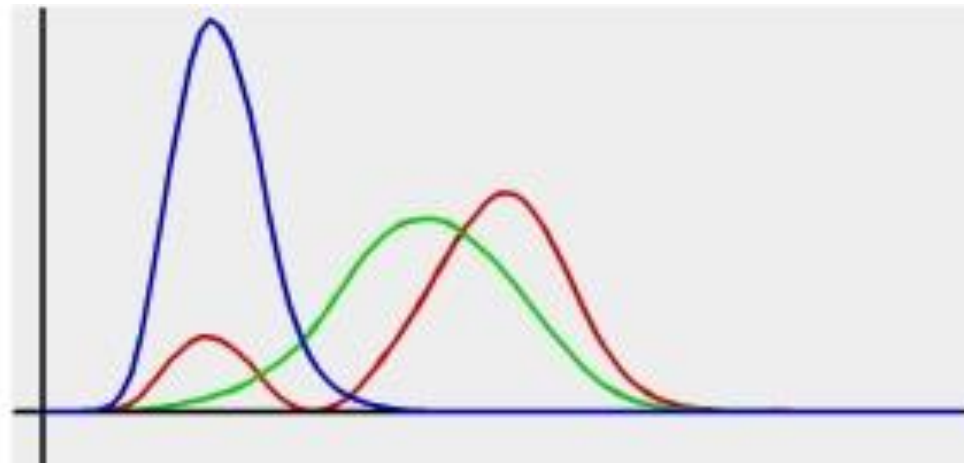
# CIE CMFS
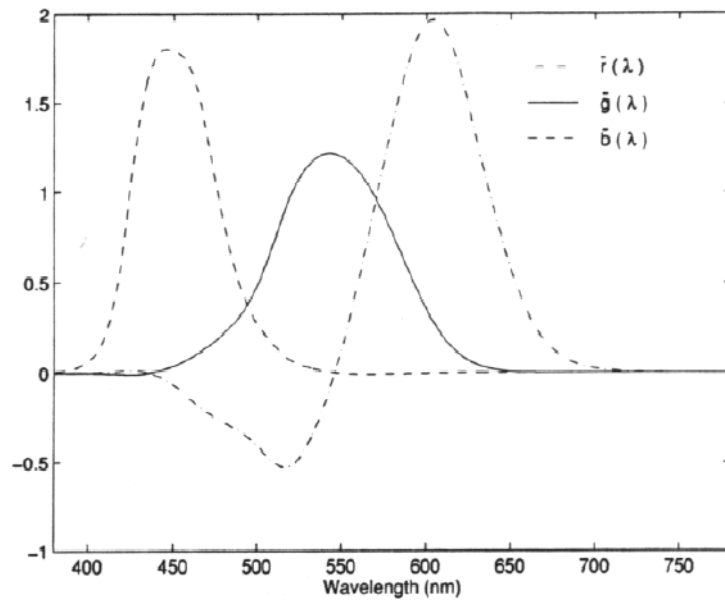


CIE RGB CMFs
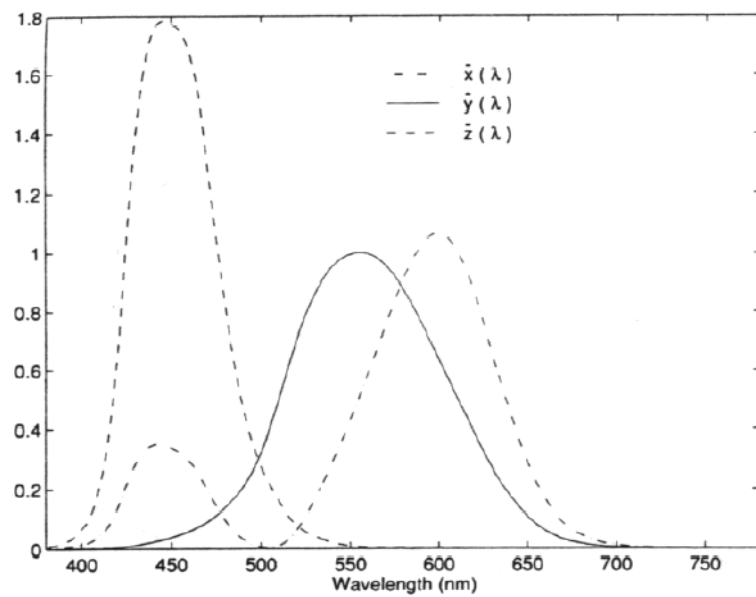
CIE XYZ CMFs

# CIE CMFS

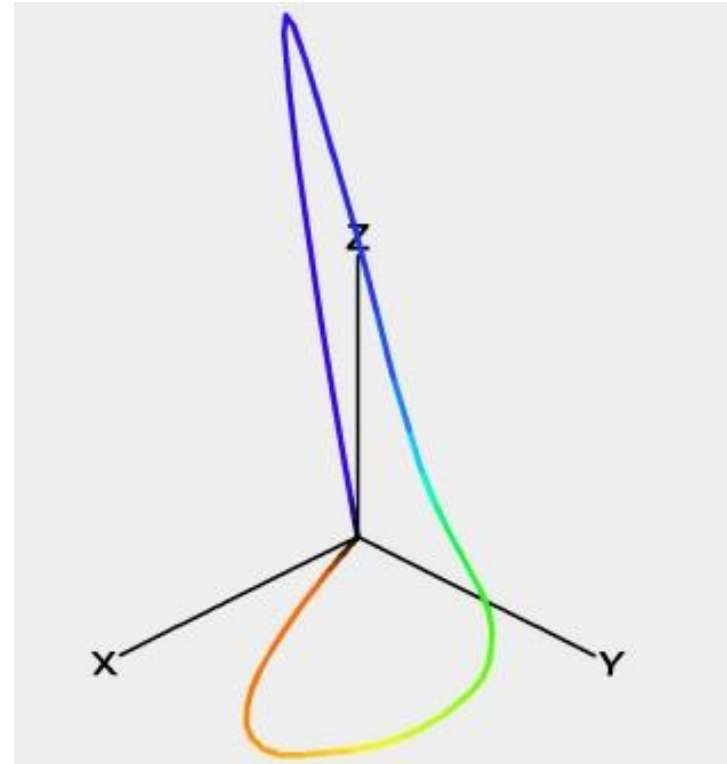**R, G, B primaries**
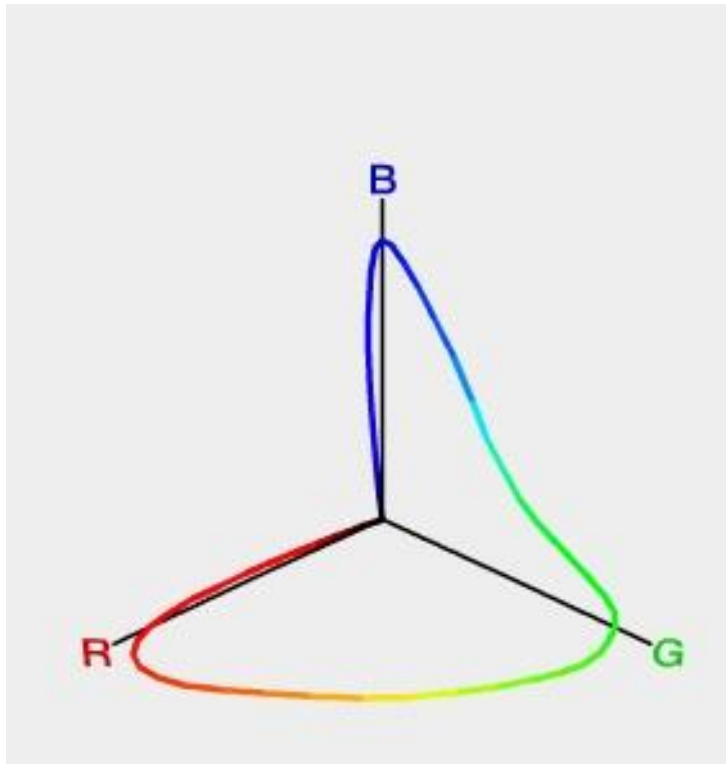(pure wavelength)

R = 615 nm
G = 525 nm
B = 445 nm

**X, Y, Z primaries**
(physically impossible but theoretically useful)

# RGB & XYZ COLOR SPACE

# CHROMATICITY DIAGRAMS

**Can we represent all colors by additive synthesis of 3 primaries?** No! **That's because CMFs can assume negative values, and the frequency responses of the color filters can only be positive.**
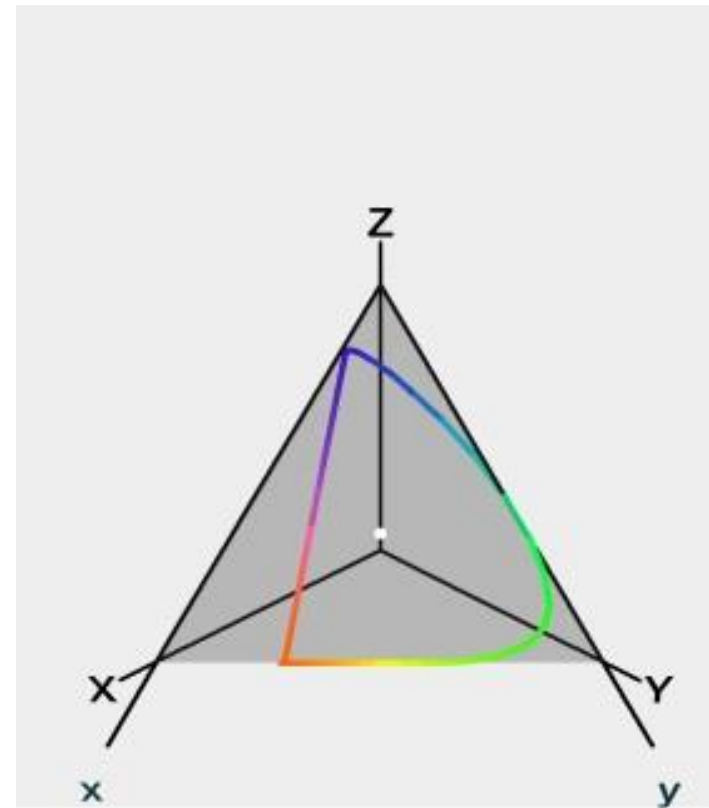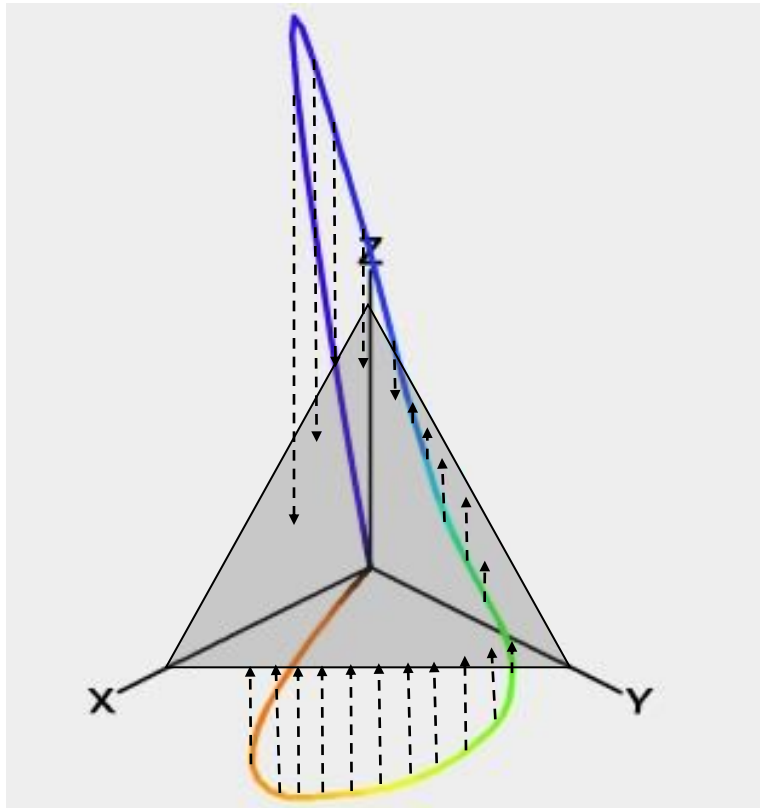
*Problem*: **represent all achievable colors on a 2-D plane (color channels** $a_1 a_2 a_3$ **live in a 3-D space!)**

*Solution*: *normalize* **colors to live in 2-D.**

*Example*: **CIE xy chromaticity diagram. Given a color value (X,Y,Z), compute the normalized values**
$$x=X/(X+Y+Z), y=Y/(X+Y+Z), z=Z/(X+Y+Z).$$

**Then, plot only the colors corresponding to 2 chromaticity coordinates (x, y)**

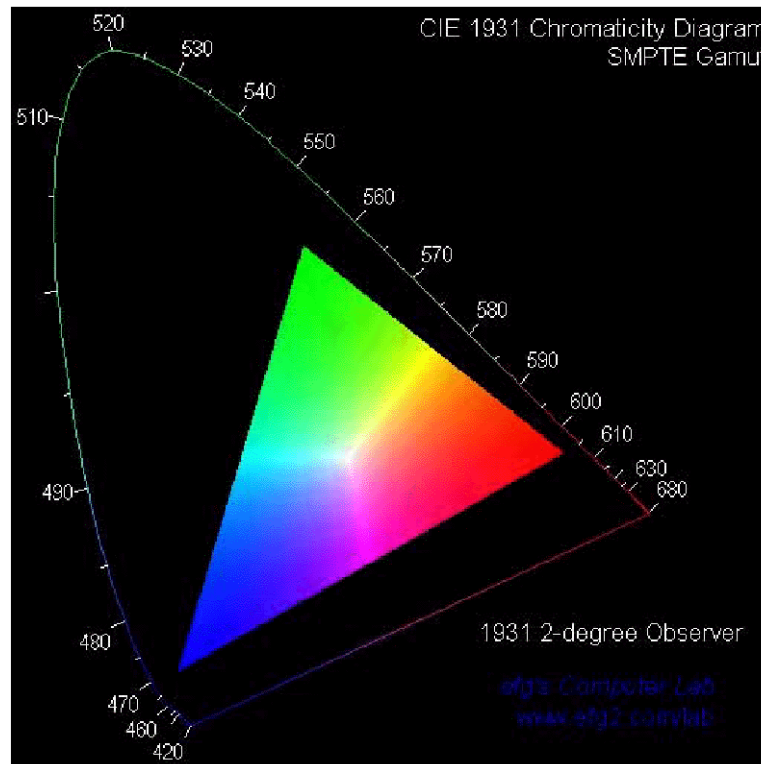# NORMALIZED XYZ SPACE

# CIE *XY* CHROMATICITY DIAGRAMS
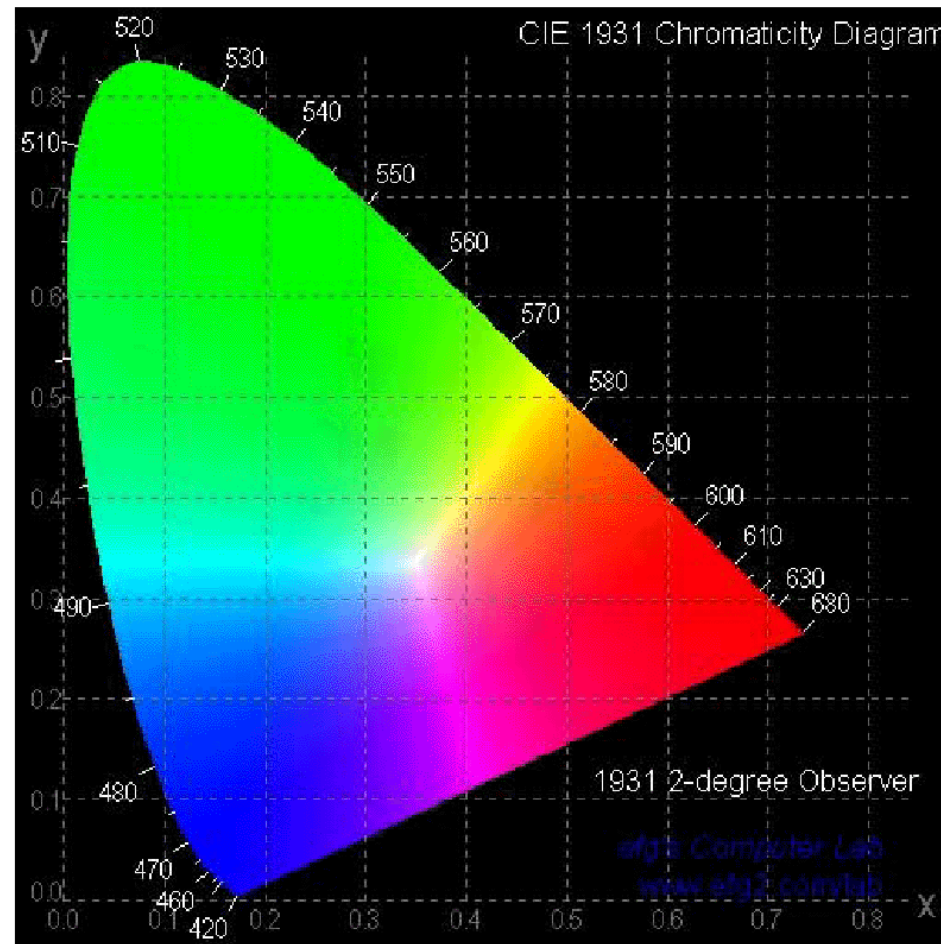
# COLOR GAMUTS

***Color gamut*** **of a color synthesis device = set of obtainable colors. Represented by a *polygonal* on the chromaticity diagram, example shown here**

# VISUALIZATIONS

**Color Applets**

**Visualization of CIE XYZ and RGB**

**Visualization of Color Gamuts**

# LINEAR TRANSFORMATIONS OF *RGB* COLOR SPACES

*RG*B: used in CRT monitors (additive synthesis)

*YIQ* (used in NTSC TV). T= $\begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & 0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix}$

*YUV* (used in PAL, SECAM). T= $\begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.1 \end{pmatrix}$

*YCrCb* (for JPEG, MPEG). T= $\begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & -0.4187 & -0.0813 \\ -0.1687 & -0.3313 & 0.5 \end{pmatrix}$

2/7/2001

# WHY IS *RGB* NOT USED?

**Similarly to the processing in the human visual system, transformed color spaces such as *YIQ, YUV, YCrCb* represent color with *luminance (*Y*)* and *chrominance* (the other 2 channels). Advantages:**

- **We can subsample the chrominance channels (e.g., 4:2:2, 4:2:0 subsampling schemes)**
- **We can quantize the chrominance channels more coarsely (with fewer bits)**
- **The chrominance channels are rather *uncorrelated* with the luminance channel, which yields better compression.**

# OTHER TRANSFORMATIONS

*CMY*: Cyan, magenta, yellow – complementary to red, green and blue respectively. Used for *subtractive synthesis* from white in color printers.

$$(C, M, Y) = (1,1,1) - (R,G,B)$$

*CMY*K: like *CMY*, uses black *(*K) as fourth color.

Given *(C, M,* Y):

K=min*(C, M, Y)*; *C=C-K; M=M-K; Y=Y-K*

# UNIFORM COLOR SPACES

**CIE XYZ is not perceptually uniform because perceptual differences between colors do not correspond to equal distance in color space**

**CIE defined two uniform color spaces:**

**CIE Luv and CIE Lab:**

- **L depends only on Y (luminance)**
- **Radial distance (e.g. sqrt($u^2 + v^2$)) correlates to chroma (or saturation)**
- **Angular position (e.g. $tan^{-1}(u/v)$) correlates to hue**

# HSV COLOR SPACE

*HSV* is based on a cylindrical coordinate system

*H* (*hue*) is measured in angles with respect to the *V* (*value*) axis. *S* (*saturation*) goes from 0 to 1.
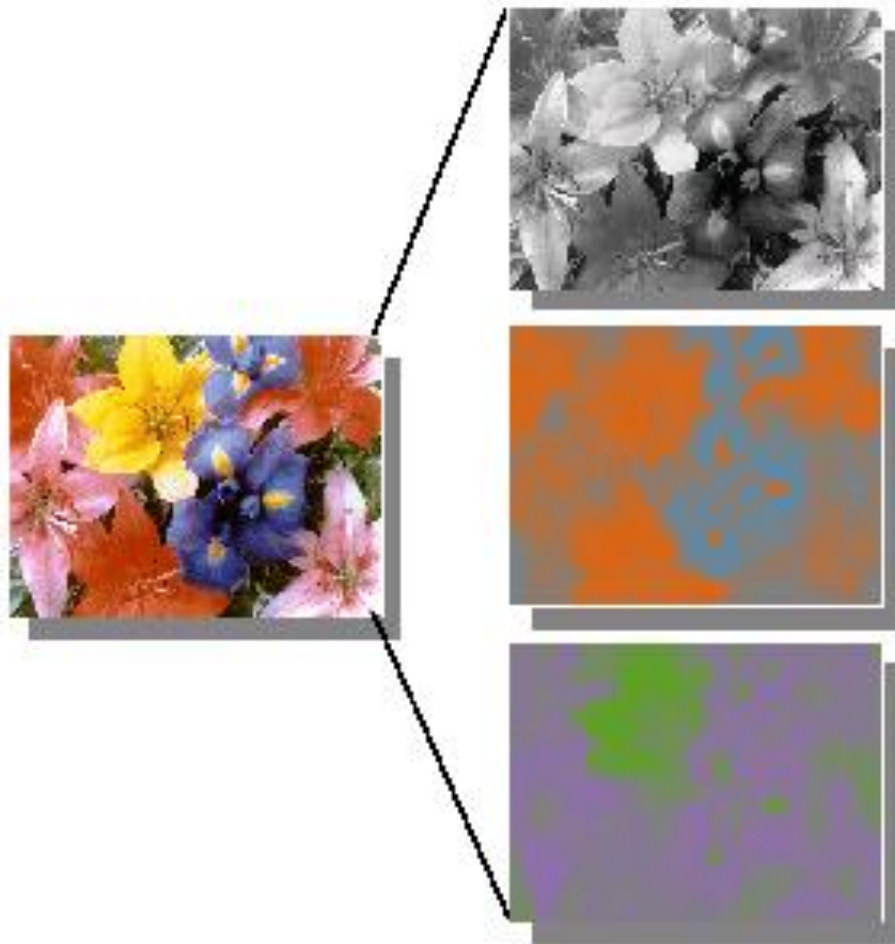
Complementary colors in HSV are 180 degrees from each other.

*Hue* distinguishes among colors such as red, green, purple, and yellow. *Saturation* refers to how far color is from a gray of equal intensity. *Value* represents the lightness (or brightness)
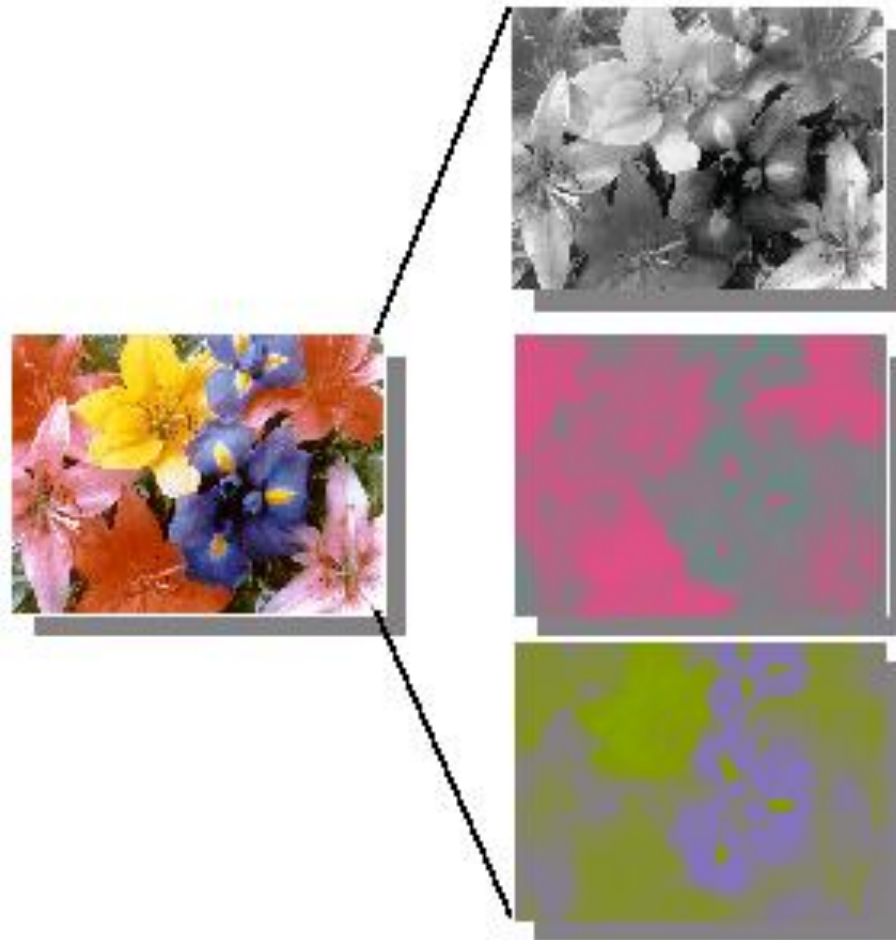
# COLOR SPACE EXAMPLES – RGB
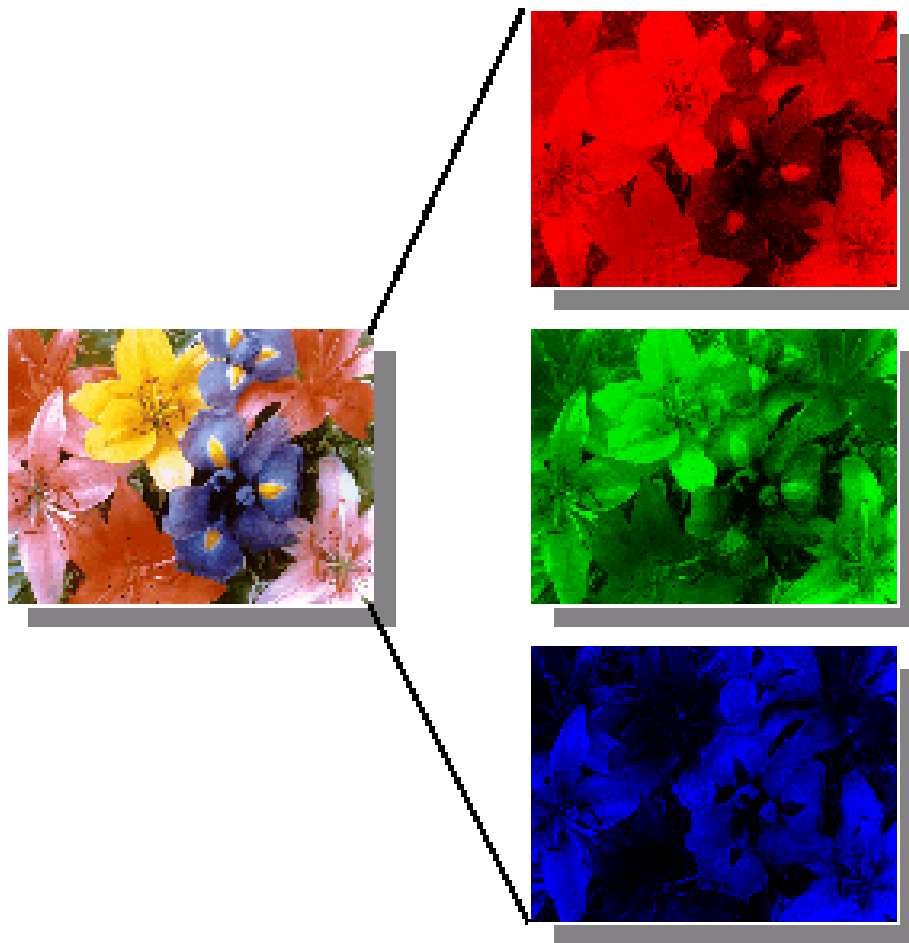
# COLOR SPACE EXAMPLES – YIQ

# COLOR SPACE EXAMPLES – YUV

# COLOR DISPLAYS

**Visible light is produce by bombardment of a thin layer of phosphor by a beam of electrons.**

**Shadow-mask color CRT: mosaic of *r,g,b* light emitting phosphors illuminated by 3 independent electron beams.**

**Color is controlled through application of different voltage to *r, g, b* guns.**

**How can we take into account non-linearities?**

# GAMMA CORRECTION

For each gun, the power of the emitted light $S_i(\lambda)$ is a function of the *control voltage* $v_i$. Ideally, the relationship would be linea*r:* $S_i(\lambda) = v_i p_i(\lambda)$ (so we would set $v_i = a_i$.) In practice, the relation is *not linear*. A better approximation is

$$S_i(\lambda) = (v_i / v)^\gamma p_i(\lambda)$$

where $v$ is the maximum value of the voltage and $\gamma$ is the exponential parameter. The monitor $\gamma$ is usually approximately equal to 2.2

Thus, given a value of $a_i$, instead of controlling the light with voltage $v_i = a_i$, we should use $v_i = v\, a_i^{1/\gamma}$

This transformation is called *gamma correction*.

# CCD CAMERAS

**3 CCD**

- **Light is refracted into each CCD using a prism**
- **In front of each CCD there is a color filter (corresponding to CMF functions *r ,g ,b)*
- **More expensive (need 3 CCDs per pixel) – but better resolution**

**1 CCD**

- **There is 1 filter *(r, g* or b) on top of each pixel. The filters are chosen according to a "mosaic" layout. E.g.**

  *r g b r g b r g b*
  *g b r g b r g b r*
  *b r g b r g b r g*

- **Cheaper (1 CCD) but with less color resolution (e.g.: for an *r* pixel, the other colors must be interpolated from nearby pixels.**

# COLOR QUANTIZATION

**What is color quantization and why is it required?**

**Frame buffer architecture: Image is stored in a video memory from which controllers constantly refresh the display screen. How many bits per pixel should we use? (8-12-16-24…)**

**The more pixels you use, the longer it takes to read and longer to refresh your monitor screen.**

**Palletized image: Uses a finite set of colors indexed in a table *(palette* or *color m*ap or LUT). You may choose this color mapping in two ways**
- **Pre designed or fixed**
- **Adaptively - "quantized" version is similar to the original**

# ADAPTIVE COLOR QUANTIZATION METHODS

The process occurs in stages
- Sampling the original image for color statistics – (Histogram)
- Choosing a color map based on those statistics
- Mapping colors to their representative colors to get new image

Different algorithms proposed and used depend on how the second step above is done
- Uniform Quantization
- Median Cut – (Heckbert)
- Octree – (Gervautz & Purgathofer)
- Peano Curve Mapping

# QUESTIONS

The chromaticity diagram in ($x$, $y$) represents the normalized color matching functions X, Y and Z. Prove that

$$Z = [ (1-x-y)/y ] Y$$

# QUESTIONS

The chromaticity space is a 2D space obtained by projecting the 3D conical XYZ space onto a plane.

- What is the equation of this plane
- Why is there no black color in the chromaticity space?
- Where do the CIE primaries lie in the chromaticity space?