Topics of this chapter

- Color Fundamentals
- Color Models
- Pseudocolor Image Processing
- Color Image Processing

Color Fundamentals

- Before: Grayscale images - pixel intensities: one value per pixel

- Now: Color images

  Each pixel has a color

  How to encode color?

Color Spectrum

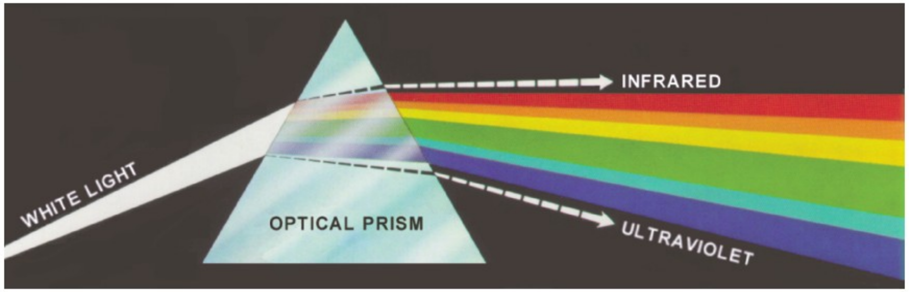


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

White light is the effect of combining the visible colors of light in equal proportions.

White light refracted in a prism reveals the color components.

Color Spectrum

Chromatic light

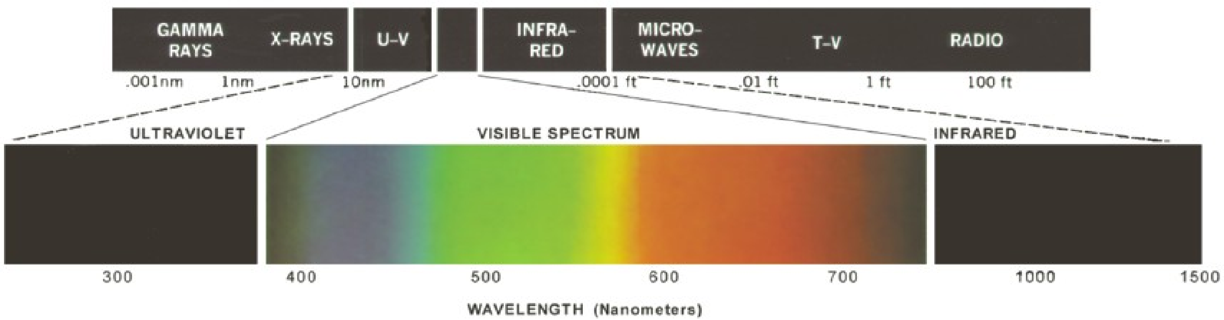


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Perceived colors are determined by the nature of the the light reflected from the object.

A body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color.

Color Spectrum

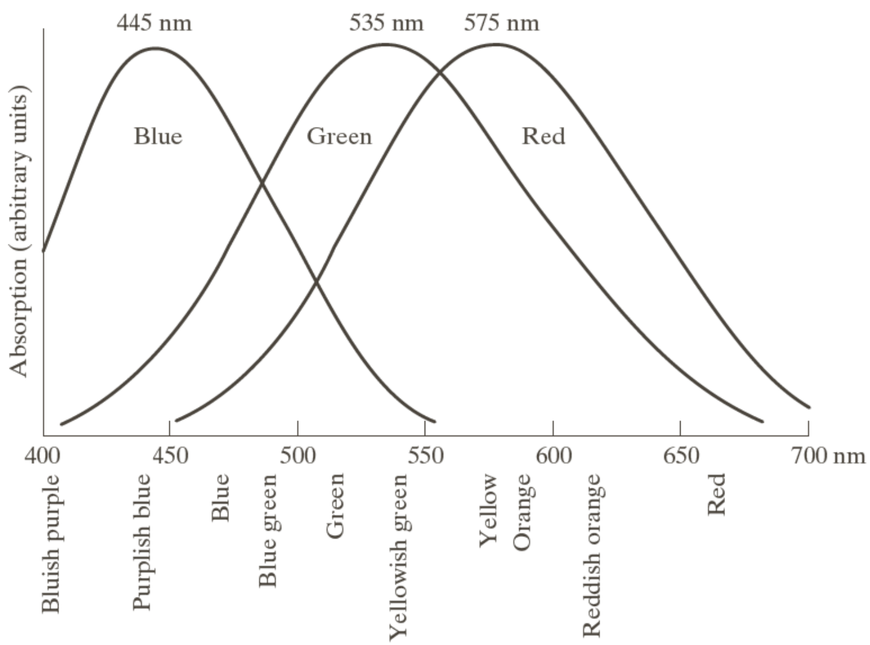
The cones are roughly sensitive to

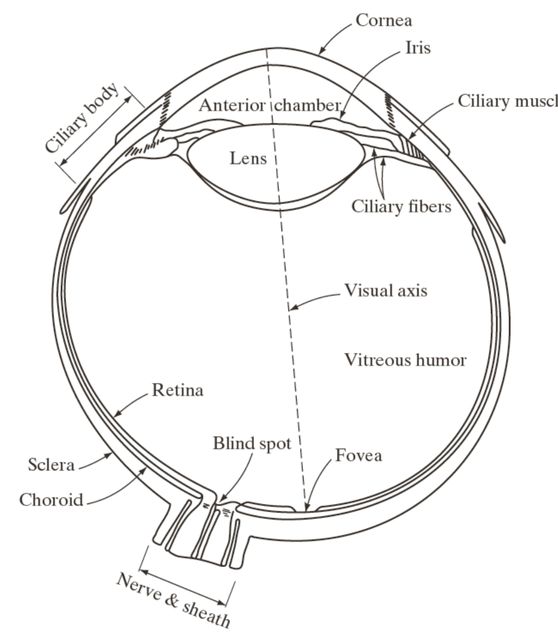
- Red (65%)
- Green (33%)
- Blue (2%)

RGB - The primary colors

3 frequency bands

FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

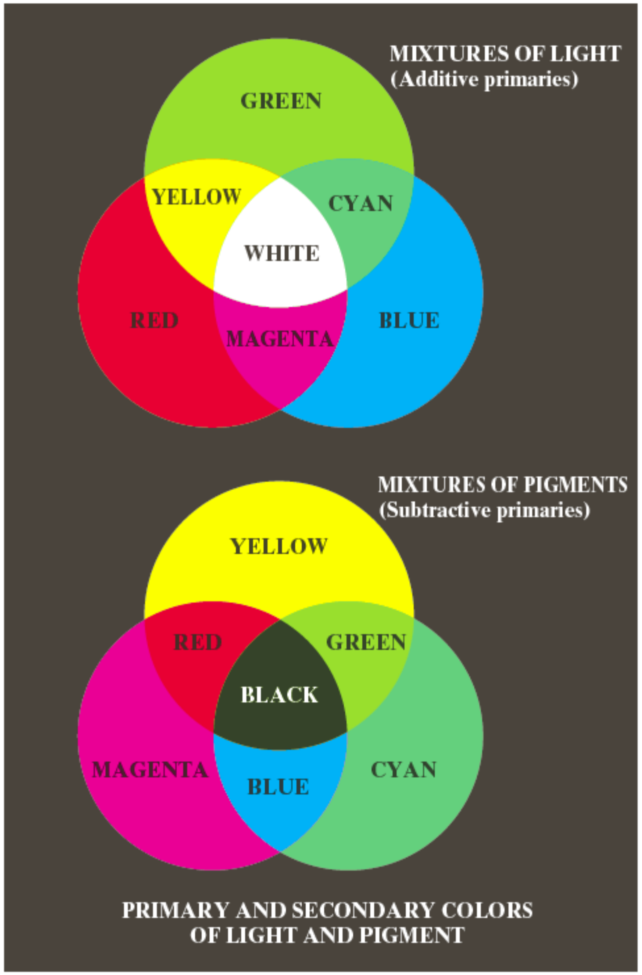




Primary and secondary colors

- R+G+B = white
- Cyan = G+B
- Magenta = R+B
- Yellow = R+G

CMY - The primary colors of pigments
substractive



Screens use RGB coding

- Electron sensitive phosphor produces light in one of the primary colors
- LEDs emit light with 3 different primary colors

Additive process

Printers use CMY+K coding

- Put pigments of different colors (absorbing light)

substractive process
K needed as C+M+Y is not really black

Trichromatic coefficients

- X: Red, Y: Green, Z: Blue

$$\left.\begin{array}{l} \bullet\ x = X\ /\ (X+Y+Z) \\ \bullet\ y = Y\ /\ (X+Y+Z) \\ \bullet\ z = Z\ /\ (X+Y+Z) \end{array}\right\} \quad \text{in } [0,1]$$

Colors are produced by adding x, y and z. Note: $x+y+z = 1$

Chromaticity diagram

- Chromaticities visible to the average person
- Function of x, y:
  $$z = 1 - (x+y)$$
- Human gamut

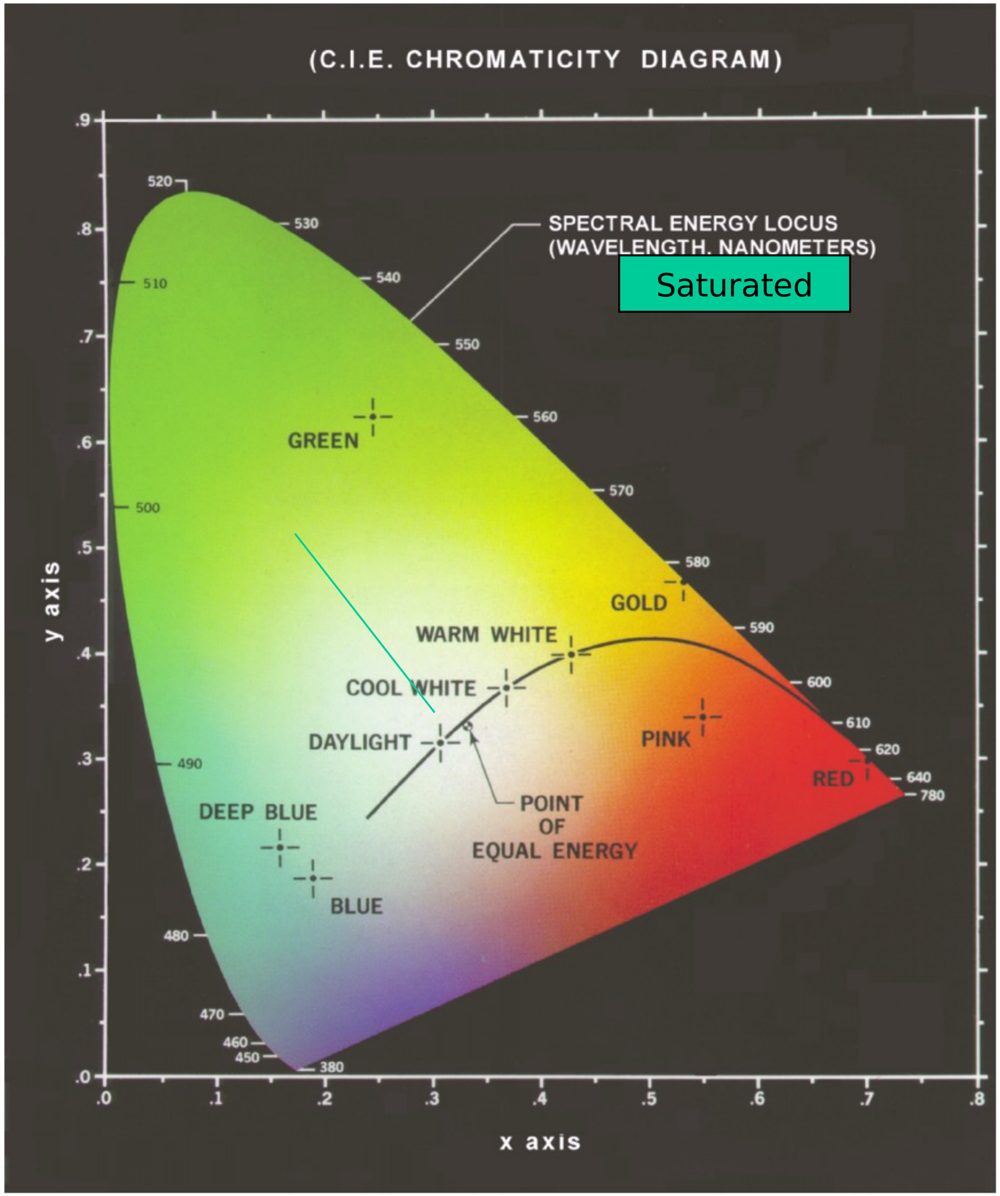
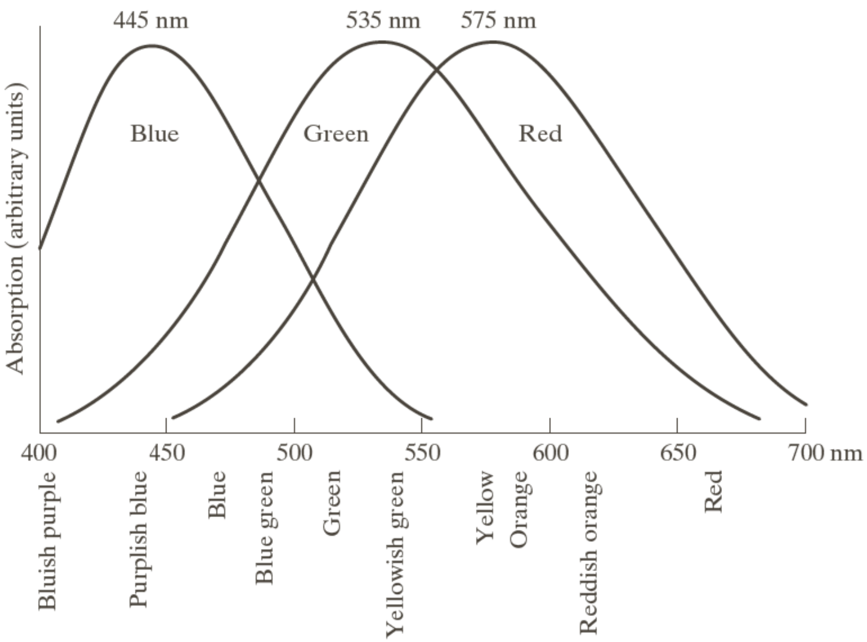
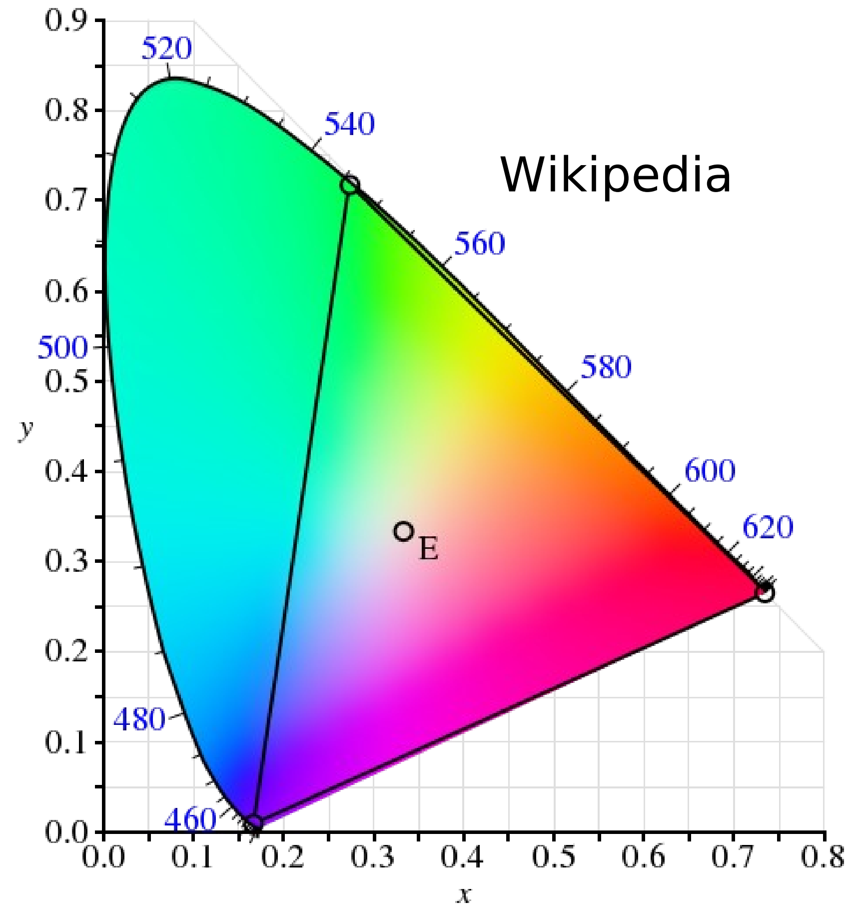


FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)

CIE standard

- Red: 700 nm
- Green: 546.1 nm
- Blue: 435.8 nm





The full human gamut cannot be covered

Chromaticity diagram

Typical

- Color gamut RGB monitor

- Color gamunt high-quality printing device

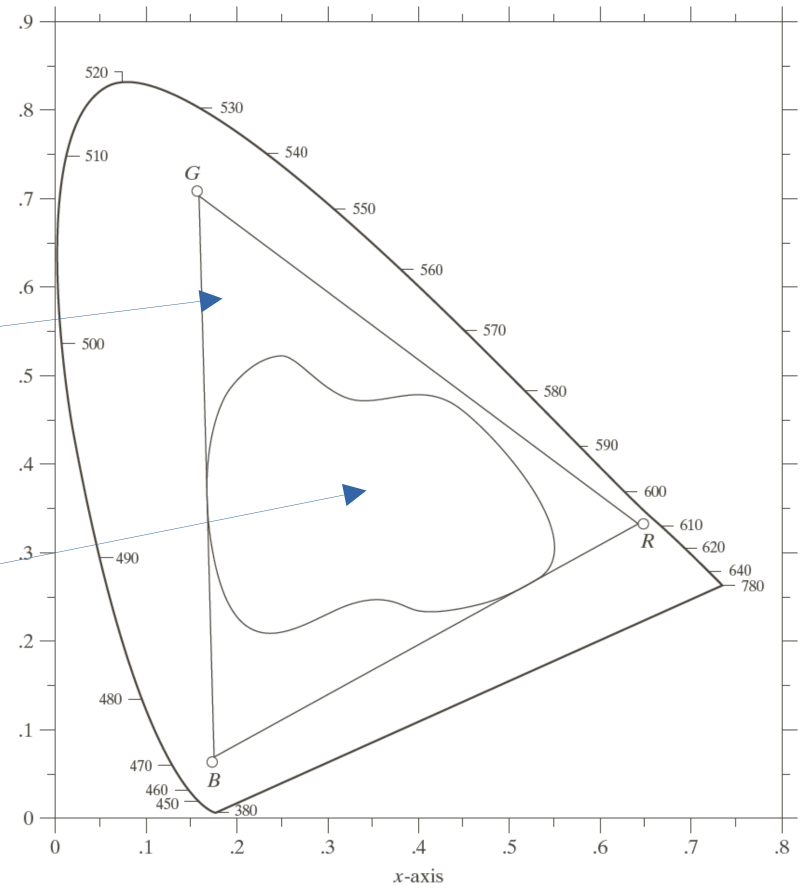


FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

Color Models

- RGB
- HSI (coming soon)
- CMY(K)

Will focus on these related color models

Purpose: To facilitate the specification of colors in some standard way

--> Coordinate systems
We need to convert back and forth

RGB Model

RGB coordinate system

- Colors as vectors (addition of x, y and z along axes)
- On or inside the cube

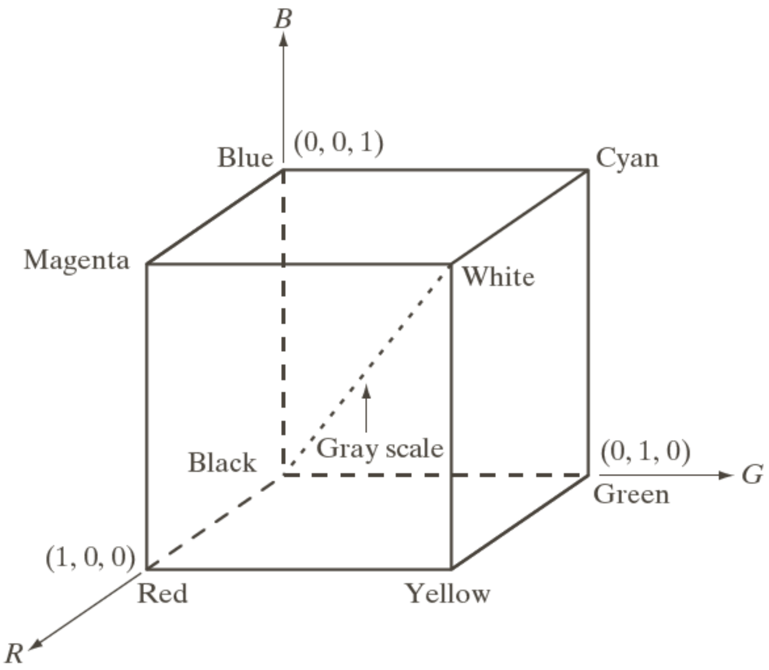
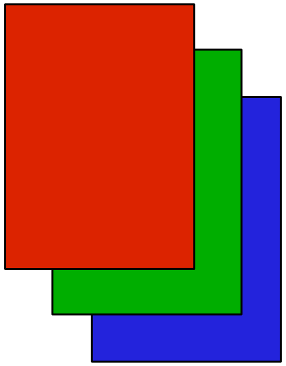


FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

Will later on relate this cube to other color models

RGB Model

RGB image planes



Truecolor image

- Assume 8 bits represent each plane (256 shades of each color)
- Number of colors: 256x256x256 > 16 mill.
- One pixel color encoded by 8+8+8 = 24 bits

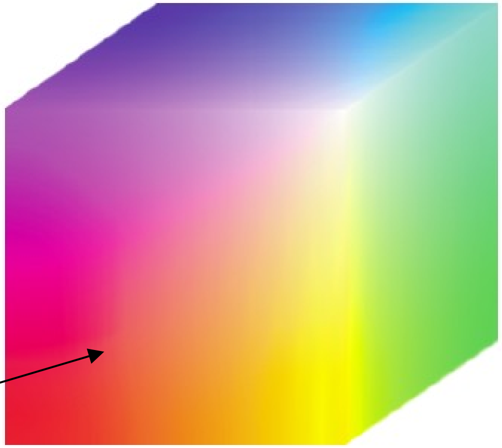


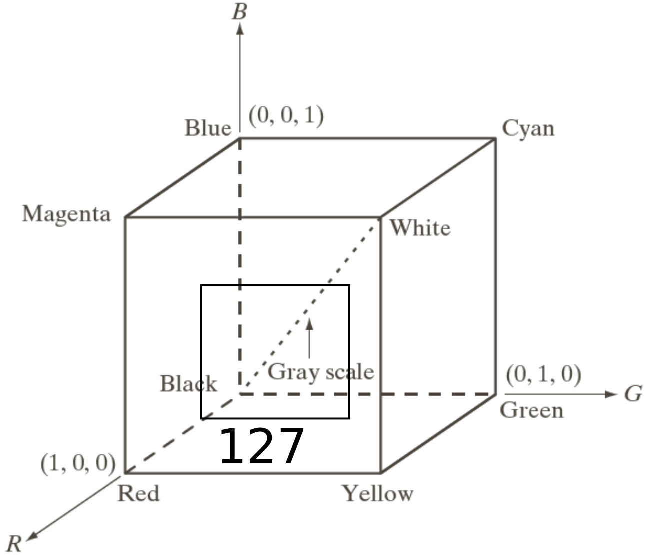
FIGURE 6.8 RGB 24-bit color cube.

One such plane is the addition of R, G and B (as in CRT)

RGB Model

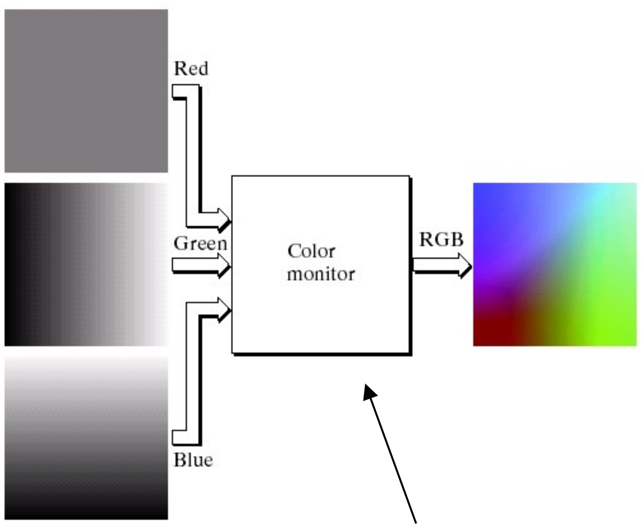
RGB image planes

- Adding RGB *intensities*



a
b

FIGURE 6.9
(a) Generating the RGB image of the cross-sectional color plane (127, *G*, *B*). (b) The three hidden surface planes in the color cube of Fig. 6.8.



Additive process

RGB intensities

RGB Model
Safe colors

Discretization of the color space: safe colors

- Reduce the memory needed 24 bits -> 8 bits
- Needed for first computers and video games

All the 216 safe colors

- Each axis has 6 steps
- Each outer plane has 36 (discrete) colors

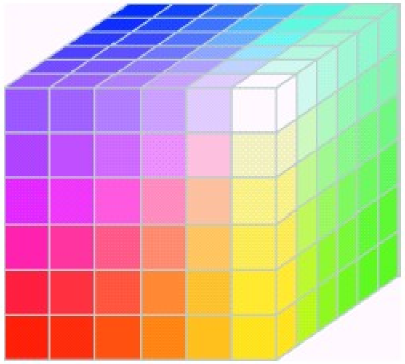


FIGURE 6.11 The RGB safe-color cube.

Modern monitors commonly 24 bits
Safe colors not so important any more

RGB Model
Safe colors

Why 216 safe colors?

- 6 shades of R, B and G, each: $6^3 = 216$
- 7*7*7=343>256

HTML colors

#FF0000

- Often sufficient
- Hex representation

Number System	Color Equivalents					
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255

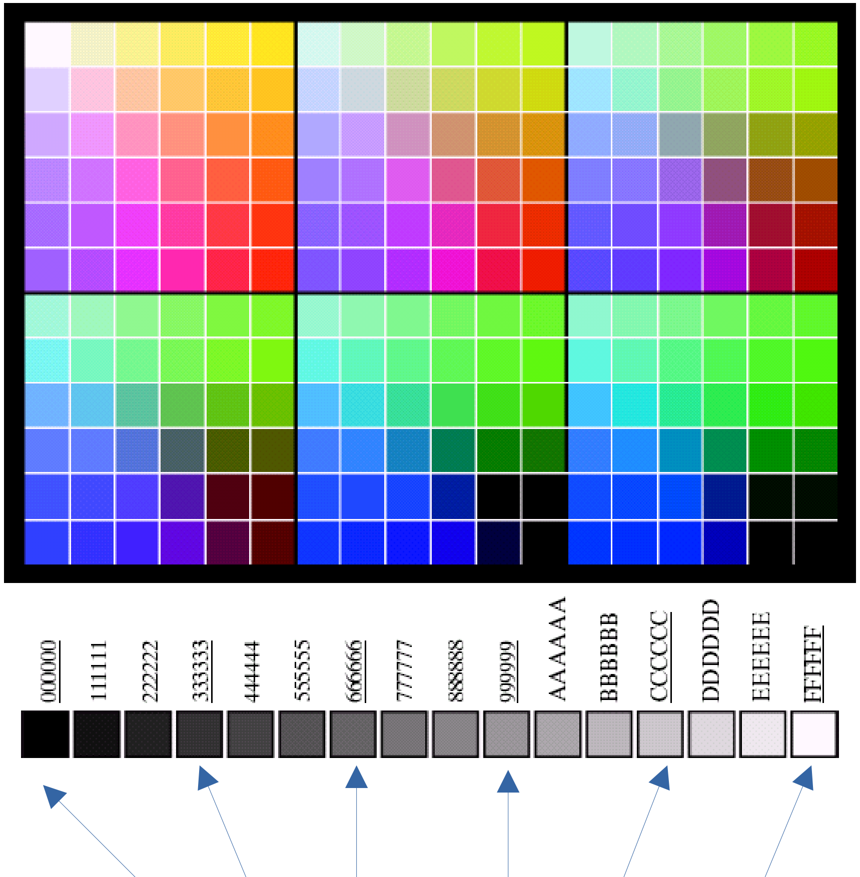
TABLE 6.1 Valid values of each RGB component in a safe color.

RGB Model
Safe colors

All the 216 safe colors

- Descending RGB values

- Not all 8-bit gray scales are safe colors



a
b

FIGURE 6.10 (a) The 216 safe RGB colors. (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

Greyscale, evenly spaced

Problem with RBG and CMY:

- Not human friendly for describing colors
    - What color is 20%R + 40% B + 40% G ?
- No simple measure of intensity
    - We would like to connect to greyscale intensity and use the image restoration methods we have seen

The HSI color model

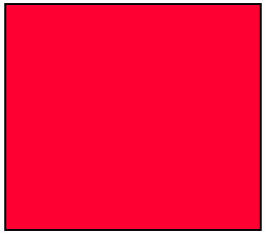
RGB are replaced by 3 other coordinates

- **Hue**: Dominant wavelength
- **Saturation**: Amount of white light mixed with hue

Chromaticity

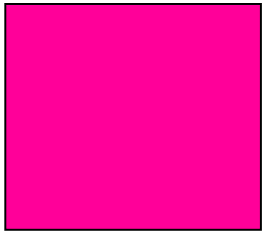
- **Intensity:** brightness

Decouples color information (H+S) from intensity



Red hue

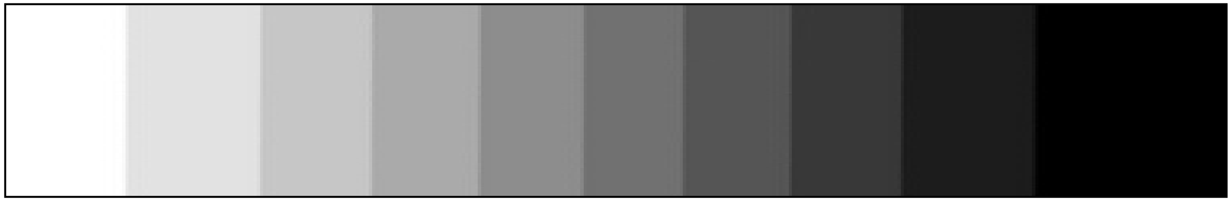
Fully saturated



Red hue

Less saturated

The HSI color model

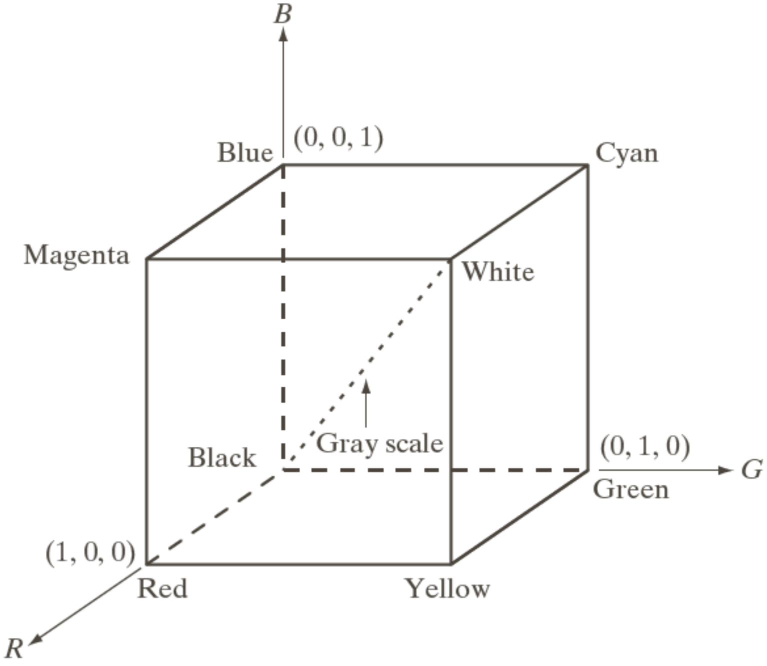


For gray scale images, intensity ranges from black to white

Gray scale intensities from colors on the line x=y=z:

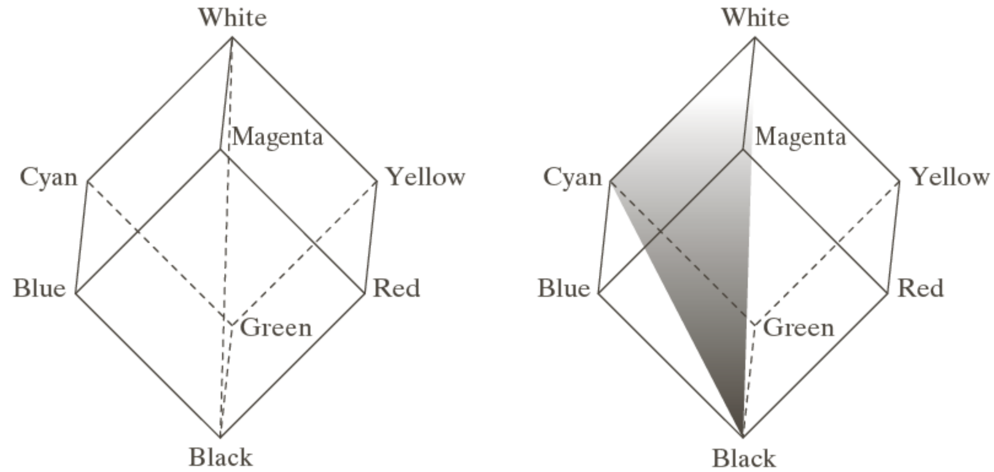
$I = (x+y+z)/3$

Color range [0,1] here



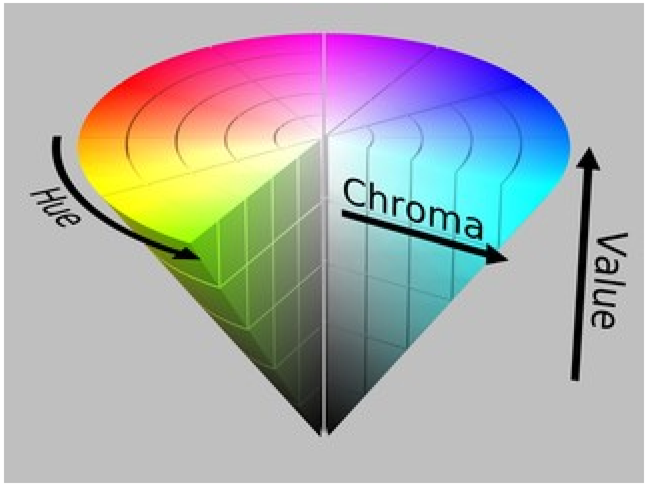
From RGB to HSI

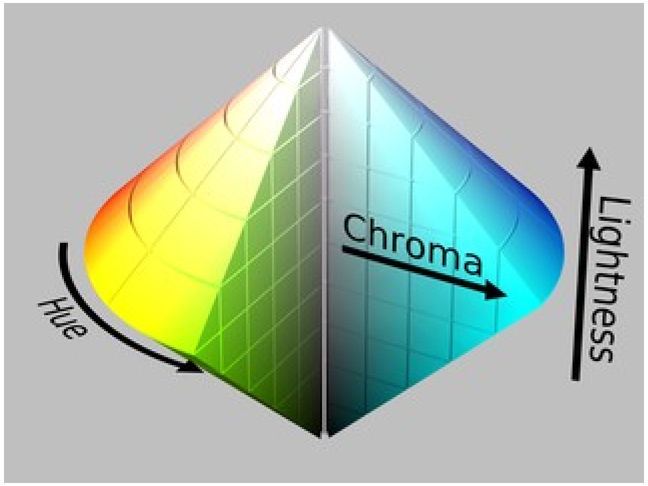
- Gray scale intensities in RGB cube: Axis from ”black” to ”white”: **Intensity**
- Chromatic info given by plane perpendicular to gray scale axis.
- Color: **Hue**
- Individual color intensity: **Saturation** or **Chroma**



a b

FIGURE 6.12 Conceptual relationships between the RGB and HSI color models.

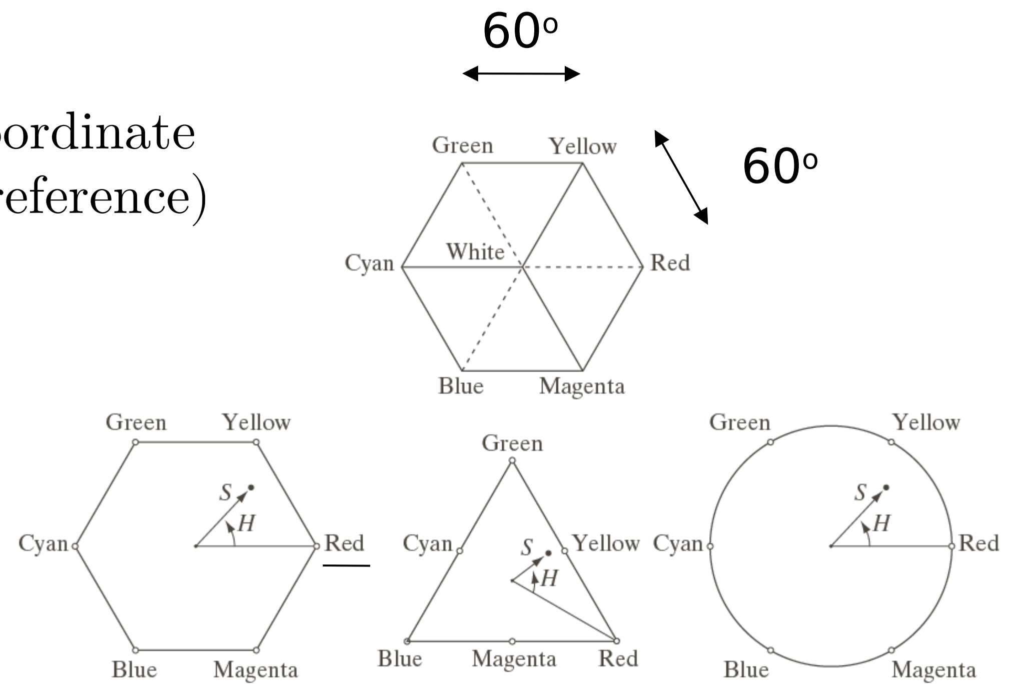




Wikipedia HSI page

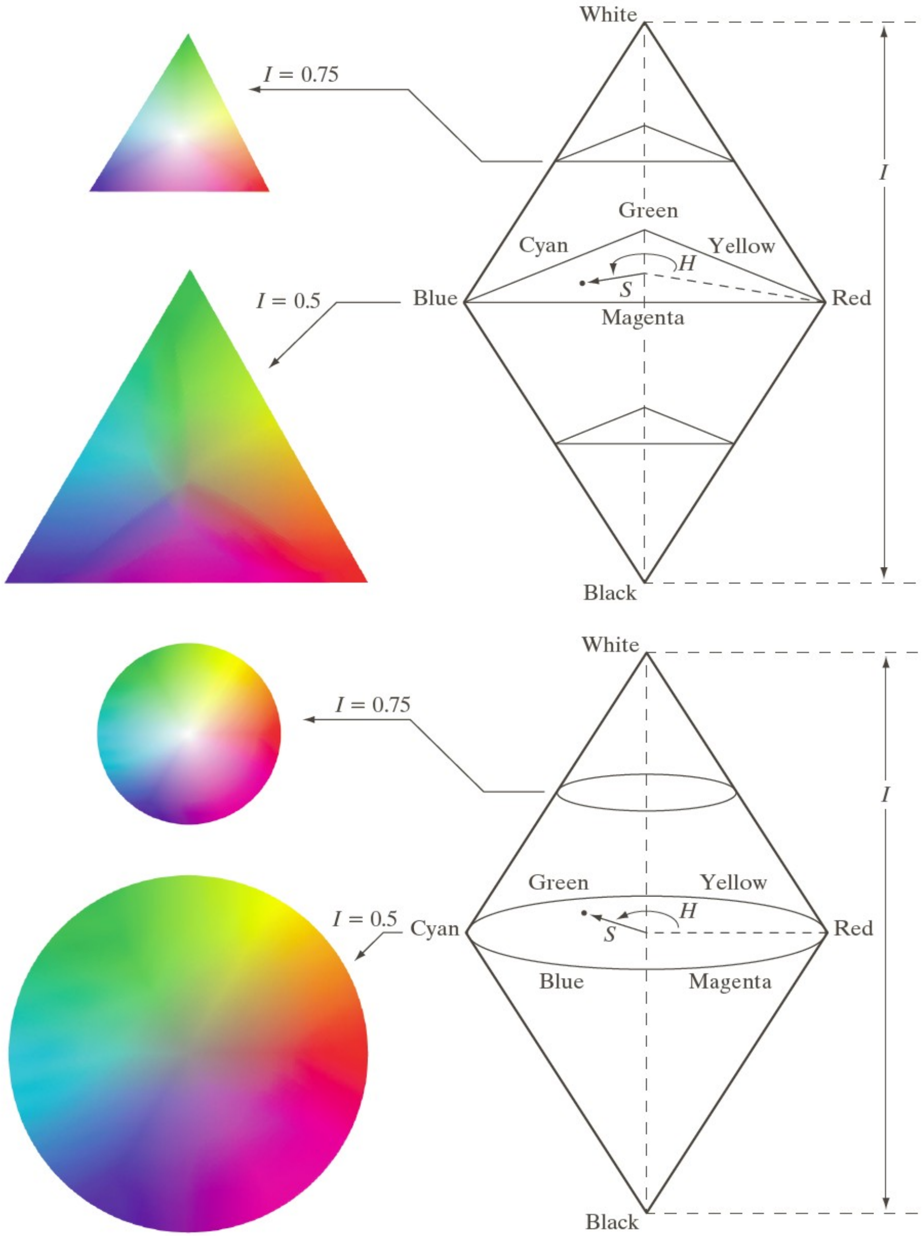
HSI
coordinate system

Chromatic coordinate
H: Hue (red reference)
S: Saturation



a
b c d

FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.



a
b

FIGURE 6.14 The HSI color model based on (a) triangular and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis.

HSI & RGB
coordinate system

RGB to HSI conversion
RGB: 3D vector (R G B)

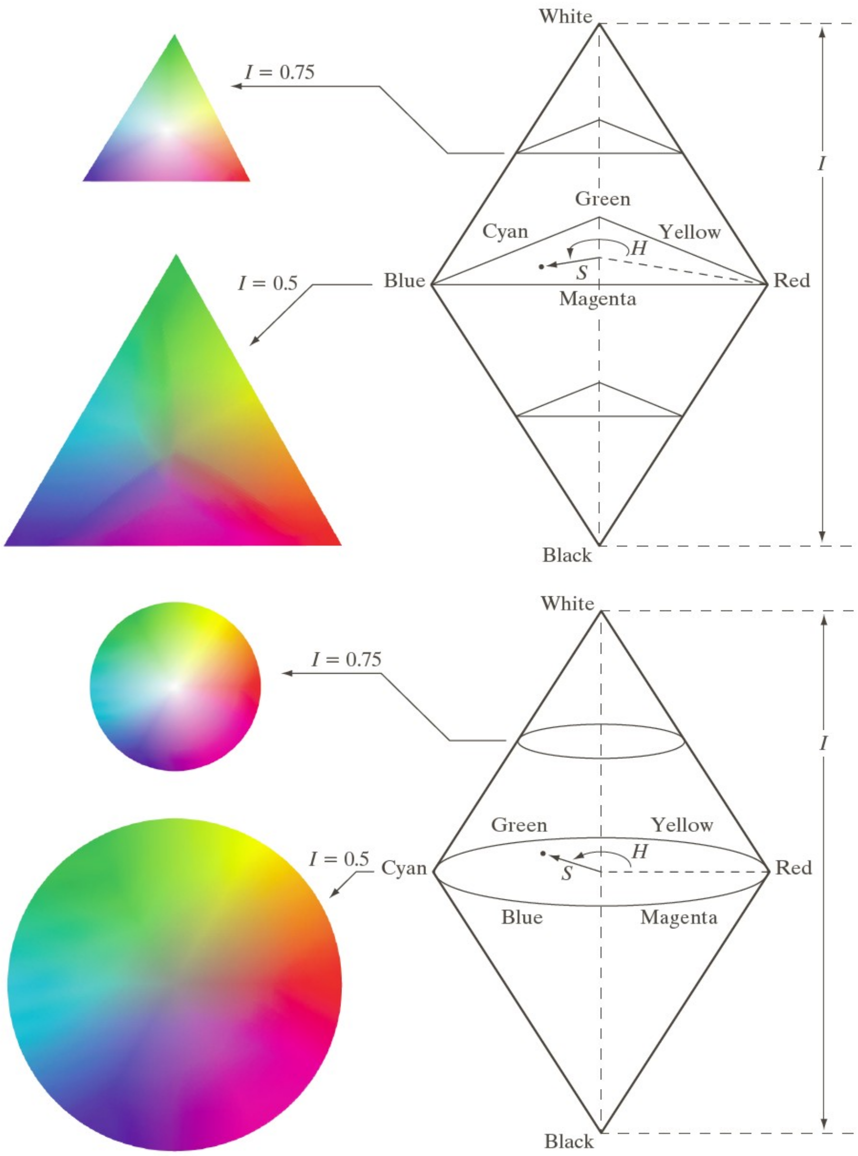
- Hue
- Saturation
- Intensity

$$H = \begin{cases} q, & B \leq G \\ 360 - q, & B > G \end{cases}$$

$$q = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \right\}$$

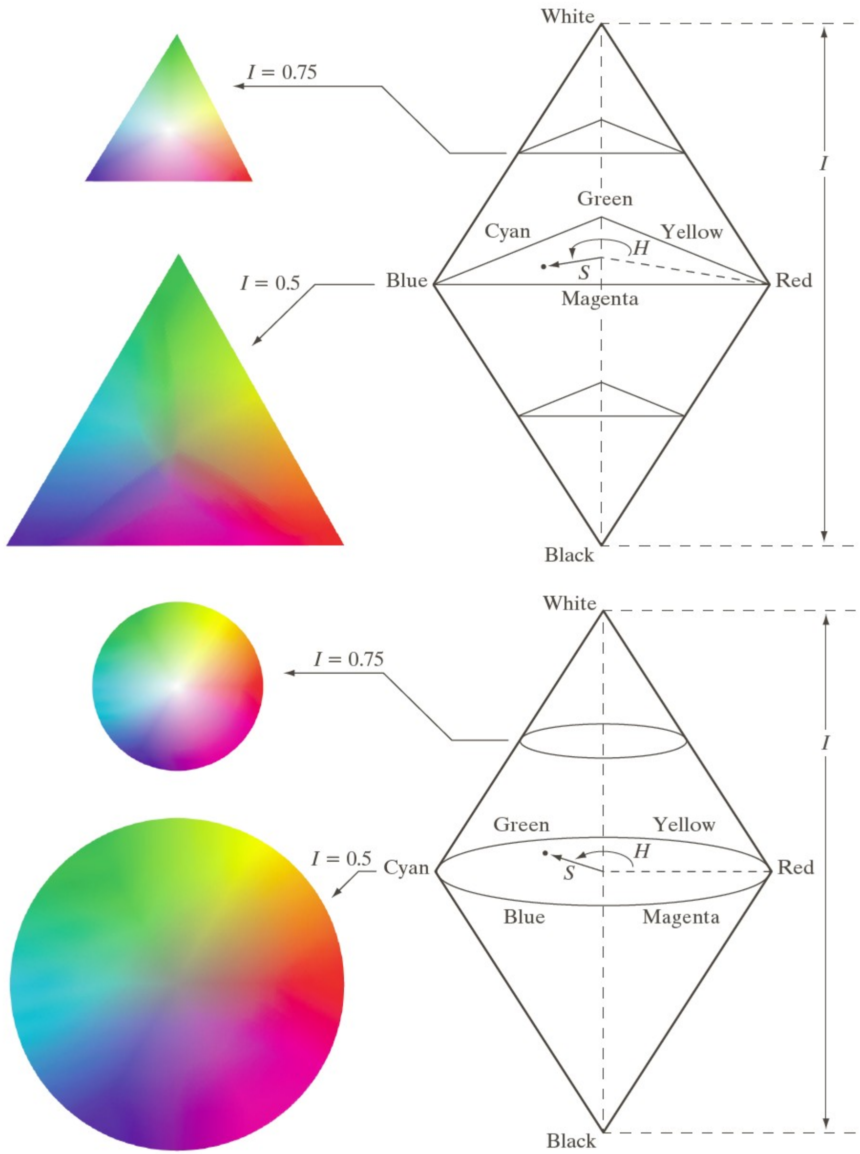
$$S = 1 - \frac{3}{R+G+B}\left[\min(R,G,B)\right]$$

$$I = \frac{1}{3}(R+G+B)$$



$$I = \frac{1}{3}(R+G+B)$$

$$S = 1 - \frac{3}{R+G+B}\left[\min(R,G,B)\right]$$

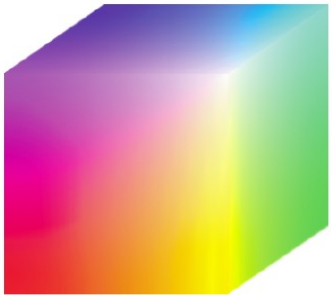


$$H=\begin{cases} q, & B \le G \\ 360-q, & B > G \end{cases}$$

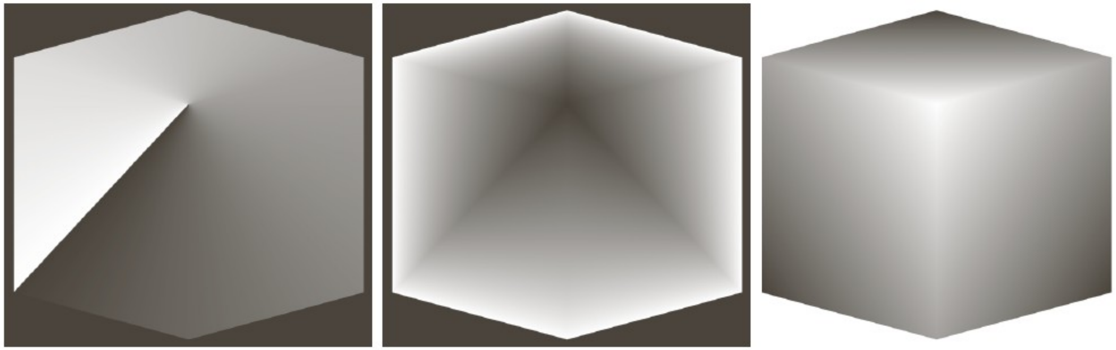
$$q=\cos^{-1}\left\{\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\}$$

$$(u, v) = \|u\| \|v\| \cos(\theta)$$

HSI Model



Example: RBG color cube to HSI



a b c

FIGURE 6.15 HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.

Discontinuity in red plane: 0-360^{o}

Less saturation toward white

Average of RGB values

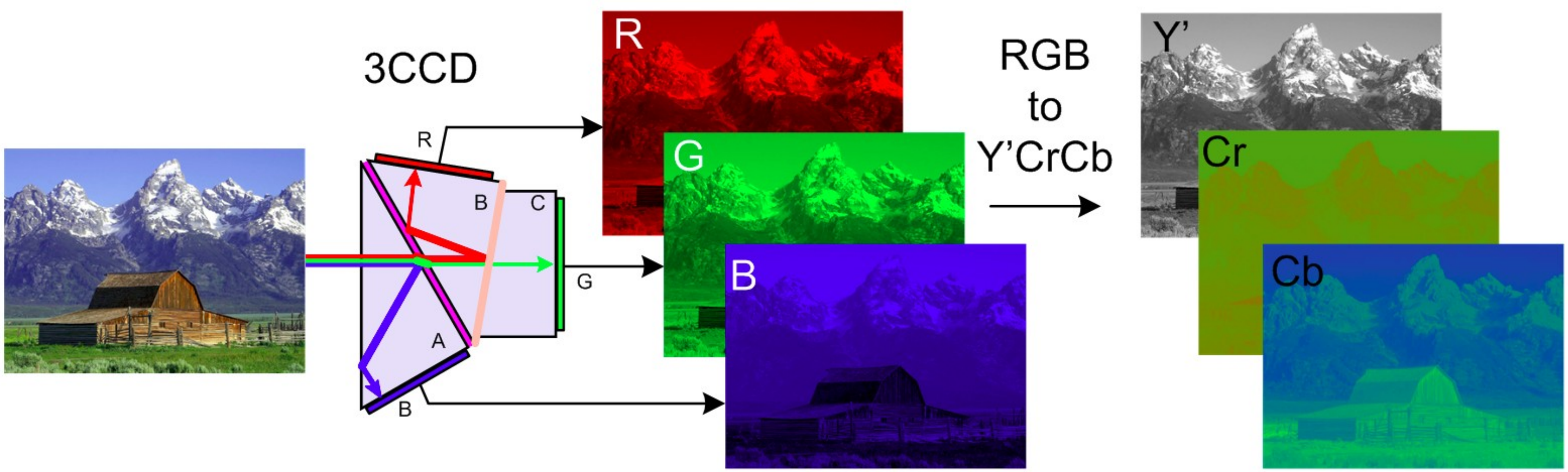
YCbCr encoding

- Used for TV (compatibility black&white – color)
- Used for image compression

$Y \sim R + G + B$
$Cb \sim Y - B$
$Cr \sim Y - R$

$R \sim Y - Cr$
$G \sim Cr + Cb - Y$
$B \sim Y - Cb$

- black&white TV uses Y only, color TV perform a simple substraction
- Used for image compression: stronger compression for CrCb



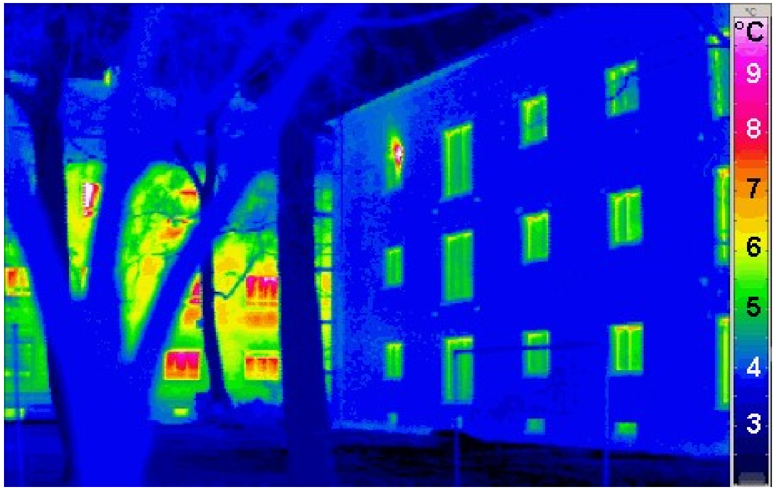
Wikipedia YCrCb page

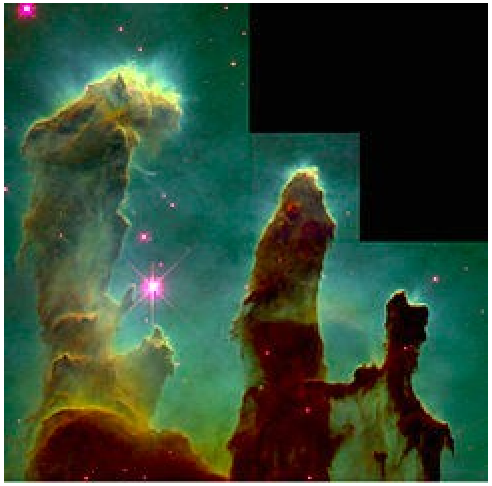
Color image processing

- Pseudo colors
- Color transformations
- Filtering with colors

Pseudocolor Image Processing

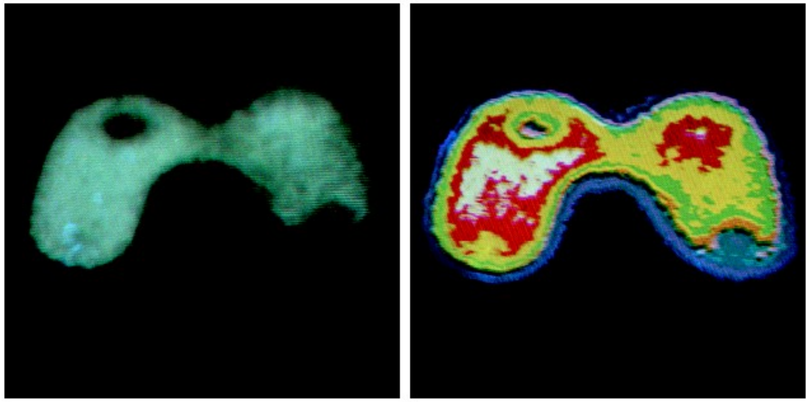
- Pseudocolor, or false color, as opposed to true color (RBG, HSI processing)
- Assigning color to gray values based on a specified criterion
- Assigning visible color to invisible color (infra-red, X-ray...)





Intensity slicing

- Slicing achromatic image into 8 color levels
- Regions appearing of constant intensity are really quite variable
- Conveys information much more easily



a b

FIGURE 6.20 (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)

- A plane is used to slice the image into two levels $f(x,y) = l_i$
- Assign different colors to each side of the plane
- Example: two-color image
- Slide plane up/down, many colors: staircase function

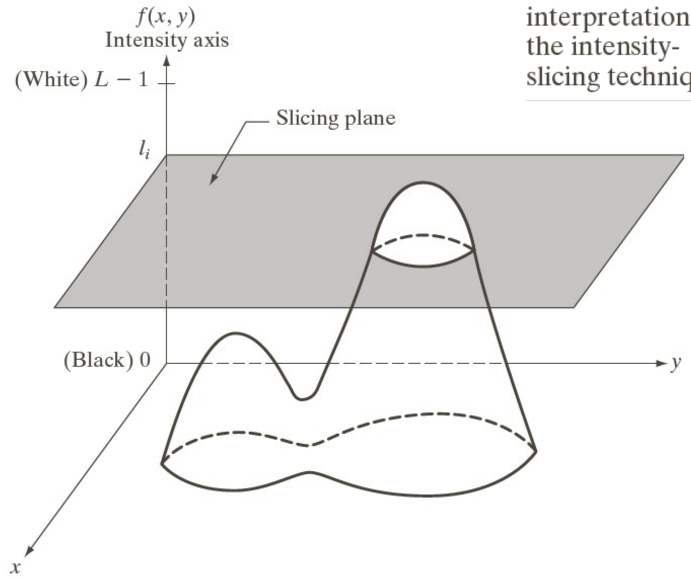


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

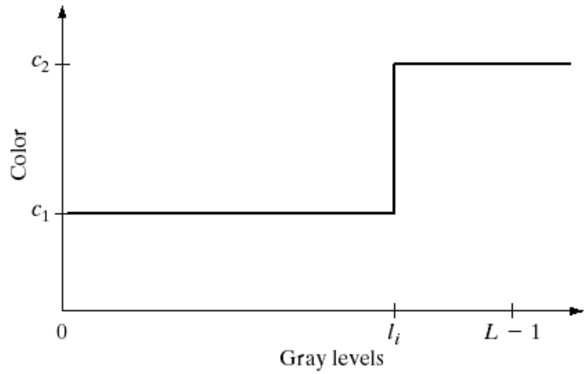


FIGURE 6.19 An alternative representation of the intensity-slicing technique.

Intensity slicing

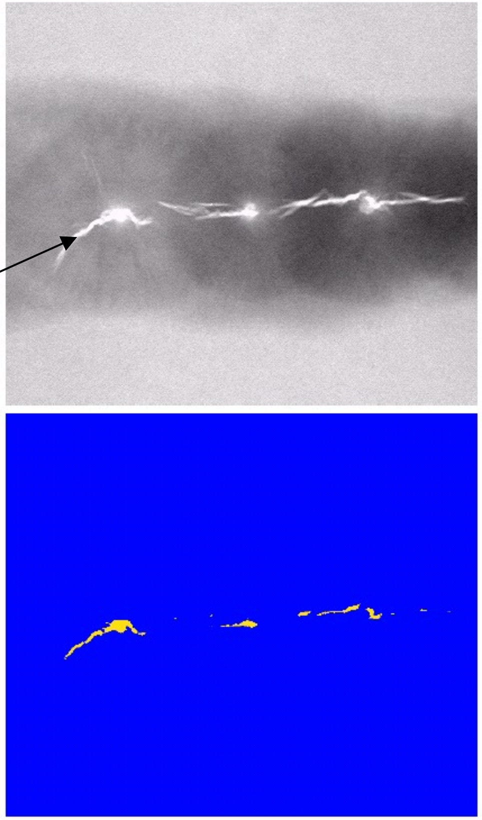
- Assigning colors according to the *physical characteristics* of the image particularly useful

- Here: X-ray imaging sensor saturates at porosities and cracks --> 255 (8 bits)

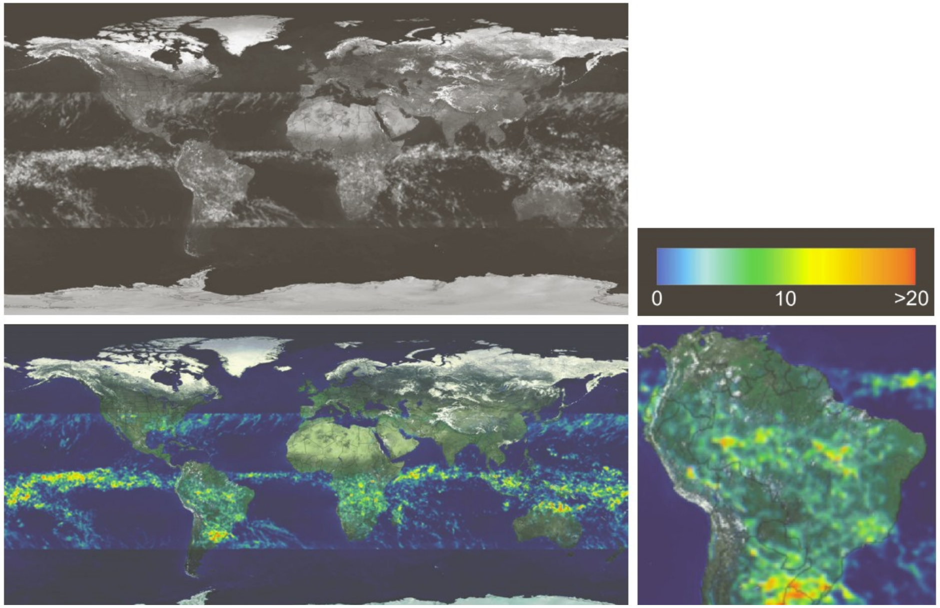
- Assign 255 to yellow, the rest to blue - aids visual inspection

a
b

FIGURE 6.21 (a) Monochrome X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)



- Many rain sensors
- Aggregated into a gray scale image with intensities corresponding to rainfall
- Difficult to grasp
- Color code: Easier



a b
c d

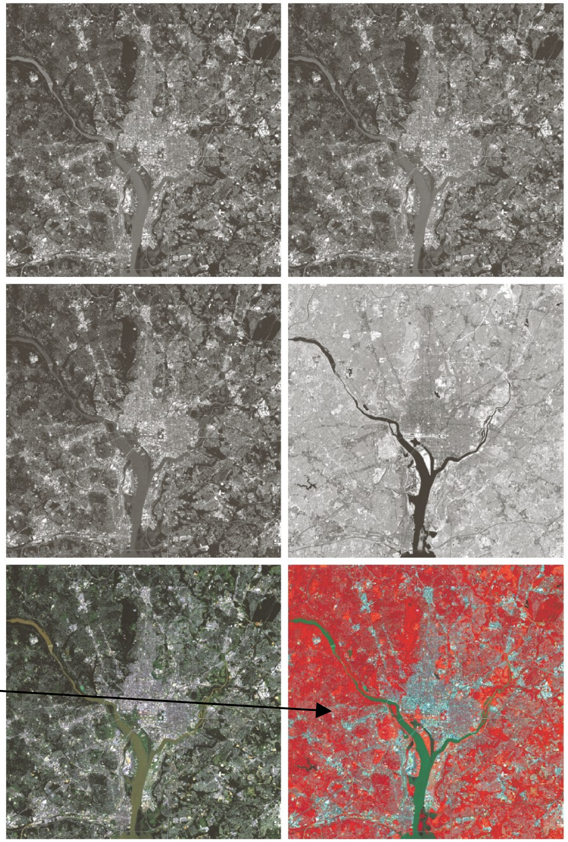
FIGURE 6.22 (a) Gray-scale image in which intensity (in the lighter horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South American region. (Courtesy of NASA.)

Color coding sensor information

- Several different sensors
- Hyperspectral images

- Produce color images e.g. By assigning different sensor intensities to RGB

Replacing red channel by near infrared

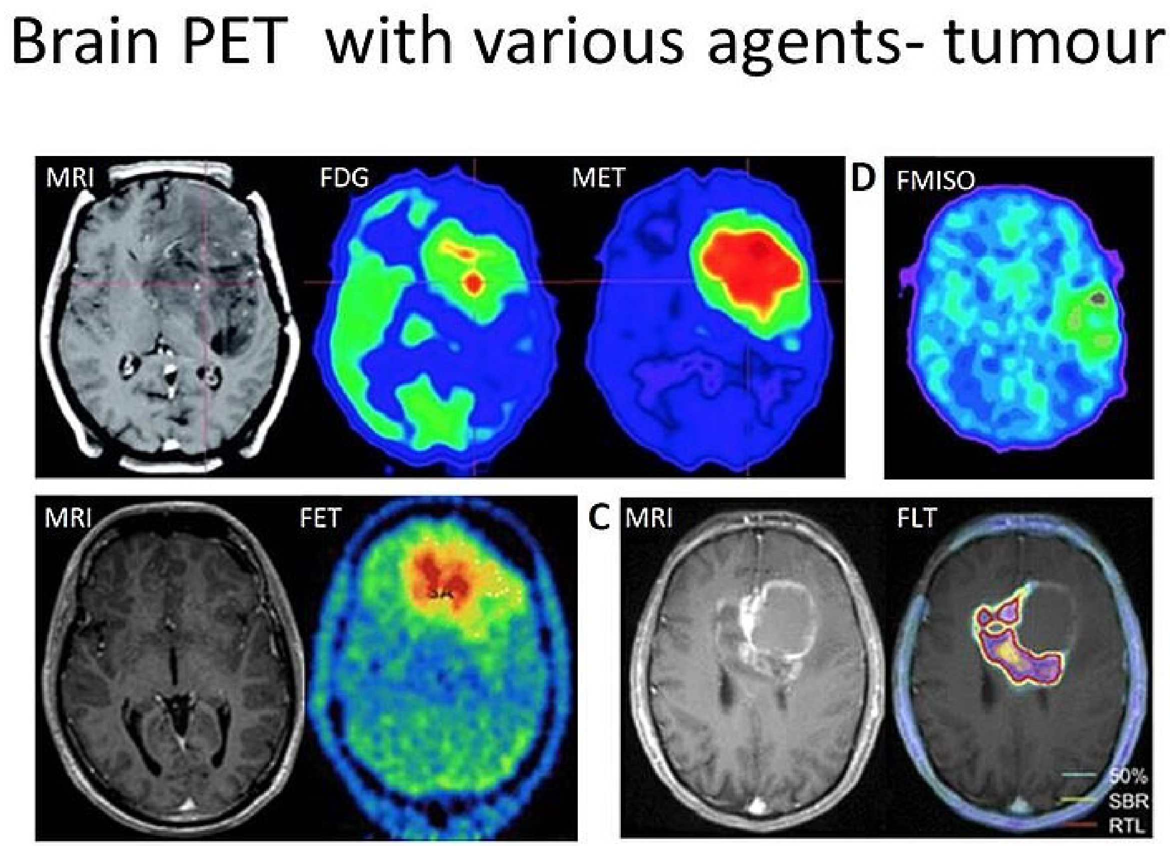


Biomass appears red

FIGURE 6.27 (a)–(d) Images in bands 1–4 in Fig. 1.10 (see Table 1.1). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (Original multispectral images courtesy of NASA.)

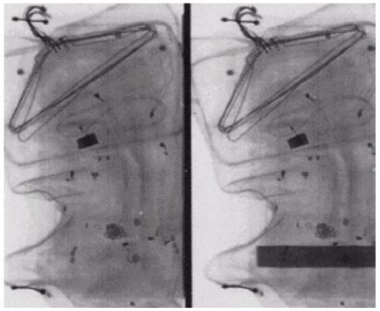
a b
c d
e f

Combining different image sources



Intensity to Color Transformations

- Airport X-ray scanning system
- Different intensities where "explosives" are placed
- Employ three different transformations



"Plastic explosives"

- Performs three independent *transformations* of the intensity of an input pixel
- Assign the three results to the RGB channels
- Color content determined by the nature of the transformations

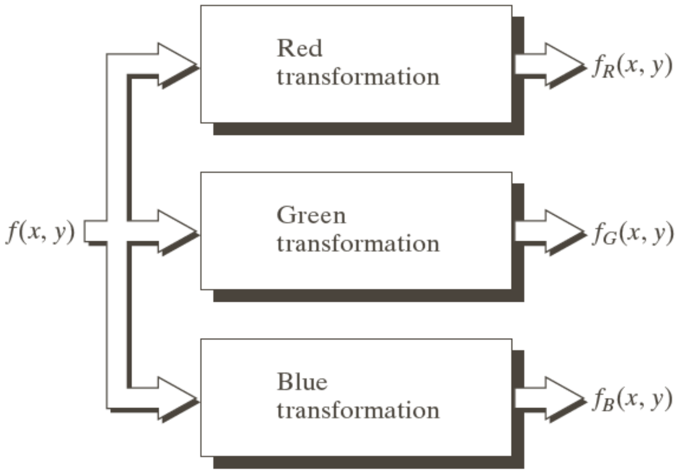
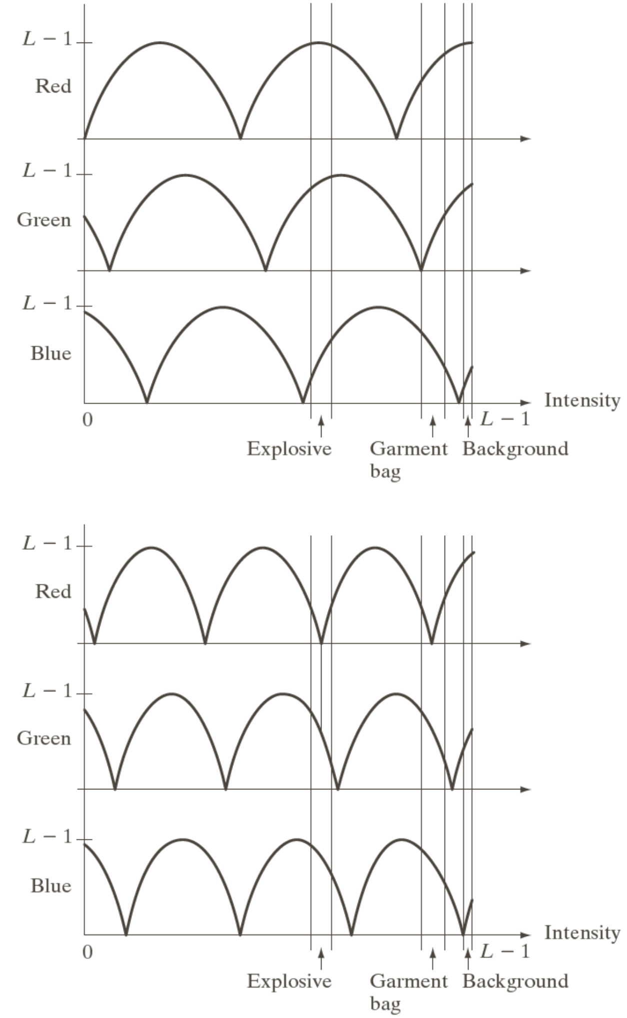


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Intensity to Color Transformations

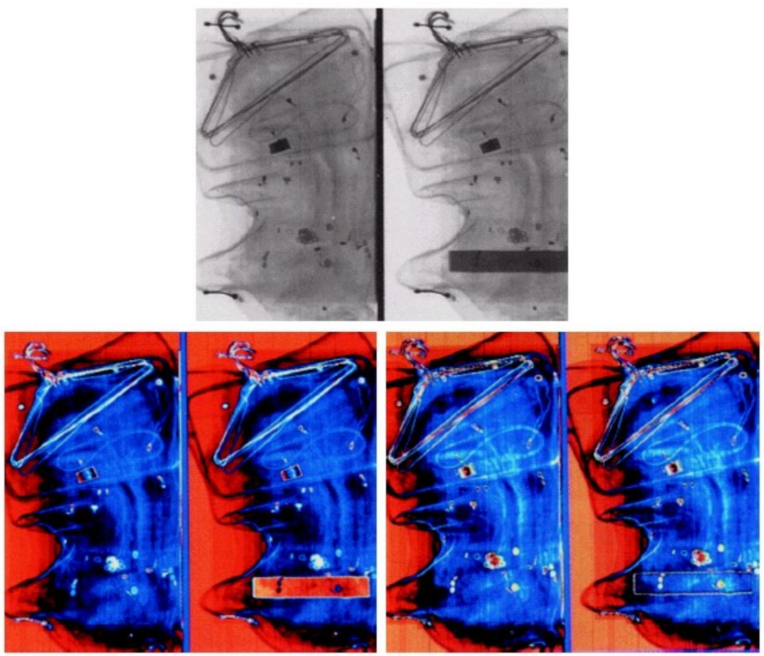
- Example: Sinusoidal functions
- Relatively constant around peaks, rapidly changing near valleys
- Changing phase and frequency can emphasize ranges in the gray scale intensities



a
b

FIGURE 6.25 Transformation functions used to obtain the images in Fig. 6.24.

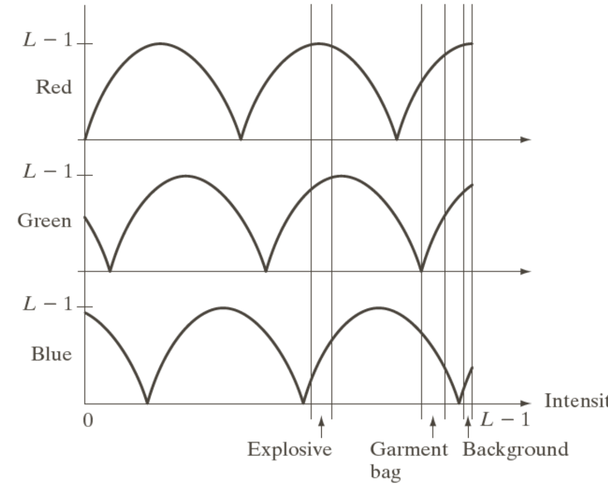
- Tune the color rendering to highlight particular intensity ranges



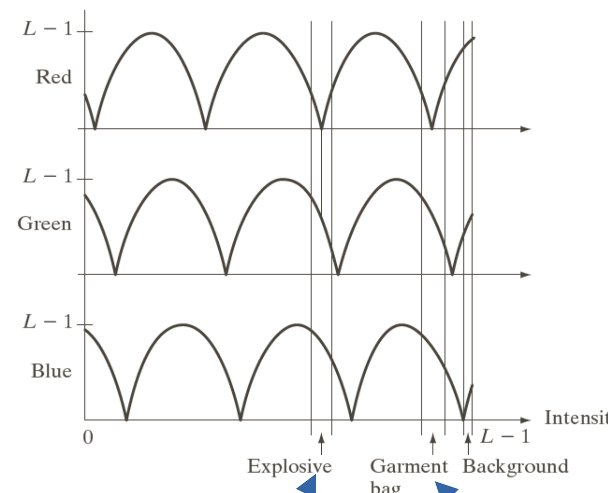
a
b c

FIGURE 6.24 Pseudocolor enhancement by using the gray-level to color transformations in Fig. 6.25. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)

b



c



Same final encoding for the explosive and bag

General framework

- Similar transformations based on several gray scale images
- Gray scale images may be obtained from sensors operating in non-visible regions

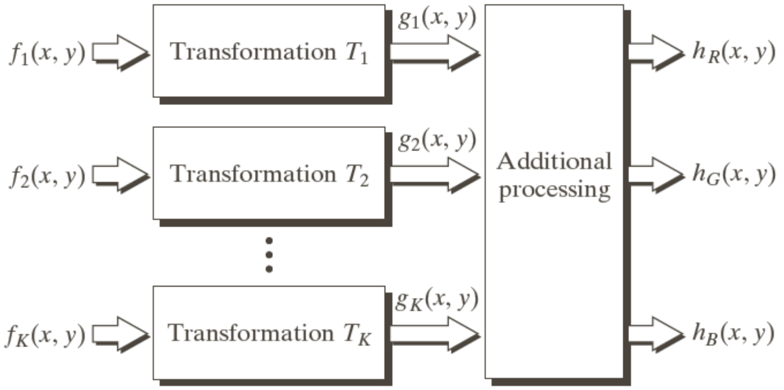
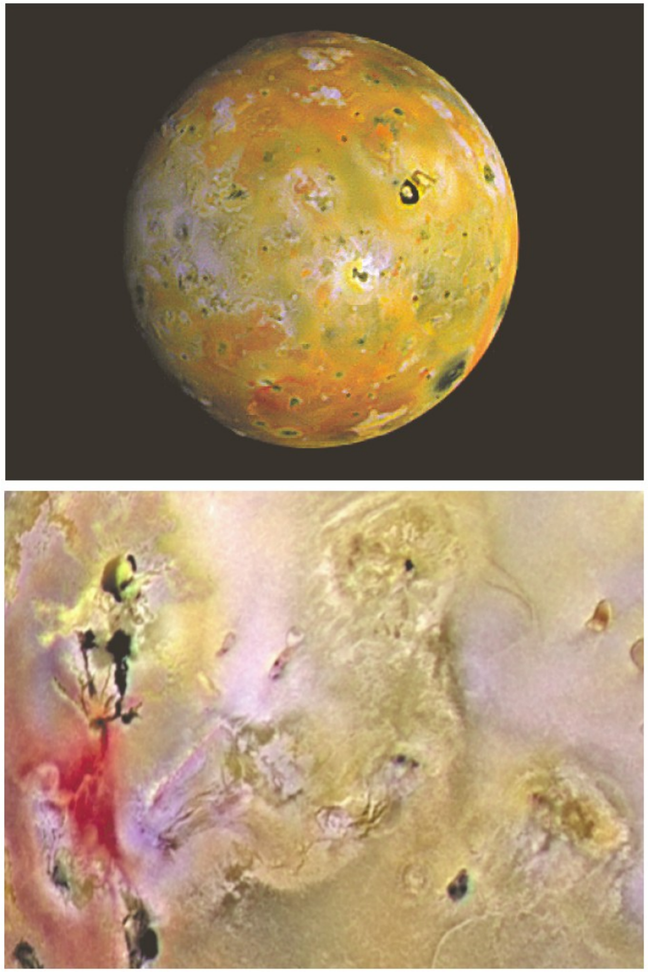


FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

Intensity to Color Transformations

- Jupiter moon Io color coded

Chemical sensors: Ejected material from volcano (red) and sulfur (yellow)



a
b

FIGURE 6.28 (a) Pseudocolor rendition of Jupiter Moon Io. (b) A close-up. (Courtesy of NASA.)

Full Color Image Processing

- Full color image processing falls into two categories
  - Componentwise processing (direct extension of previous methods, each color gives a grayscale image), then form resulting color image
  - Process color pixels directly. These are *vectors*

Example: RGB

$$f(x,y)=c(x,y)=\begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

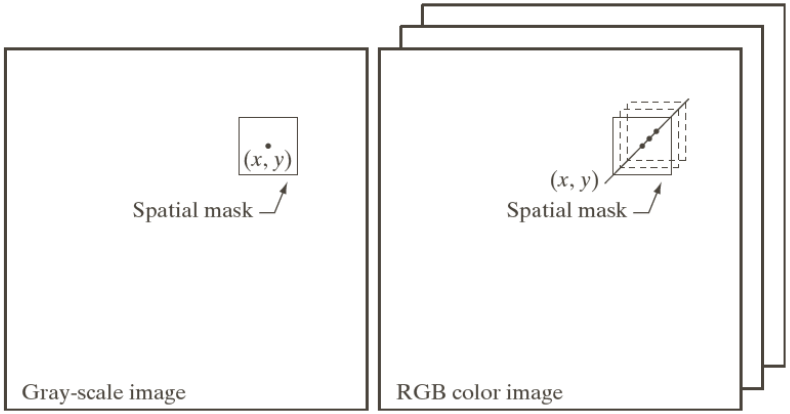
- Per-color and vector-based processing are equivalent if

  - The process is applicable to both vectors and scalars (e.g finding a mean)

  - *Independent* componentwise processing

we do not want to change the original colors

$$I = \frac{1}{3}(R + G + B)$$



a b

FIGURE 6.29 Spatial masks for gray-scale and RGB color images.

Simple Color Transformations

- In general $g(x,y) = T[f(x,y)]$
- Vectors. E.g. RGB
  $$s_i = T_i(r_R, r_G, r_B),\ i = R,G,B$$
- One output color component may depend on all input color components

FIGURE 6.30 A full-color image and its various color-space components. Interactive.)

- Example: Modify the intensity
  $g(x,y) = kf(x,y) \quad \text{where } 0<k<1$

- RGB:

$$kc(x,y)=k\begin{bmatrix}R(x,y)\\G(x,y)\\B(x,y)\end{bmatrix}=\begin{bmatrix}kR(x,y)\\kG(x,y)\\kB(x,y)\end{bmatrix}$$

- Componentwise

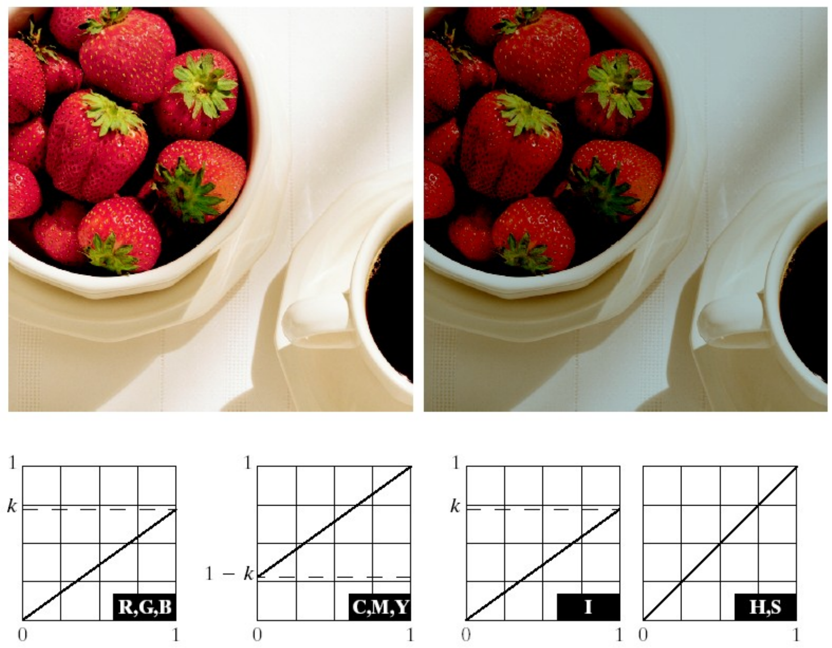
$$I=\frac{1}{3}(R+G+B)$$

Input image

Simple Color Transformations

- Example: Modify the intensity
  $$g(x,y) = kf(x,y)$$
  where $0<k<1$

- For HSI, intensity is decoupled from color (HS) $s_I = kr_I$
- Does not change the color, only the intensity



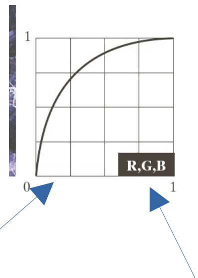
a b
c d e

FIGURE 6.31 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)

Per-color-component basis

Simple Color Transformations

- Tonal correction, gamma correction. Same function for all 3 colors
- May change the color!



Small values are increased more than high values: color mix can change

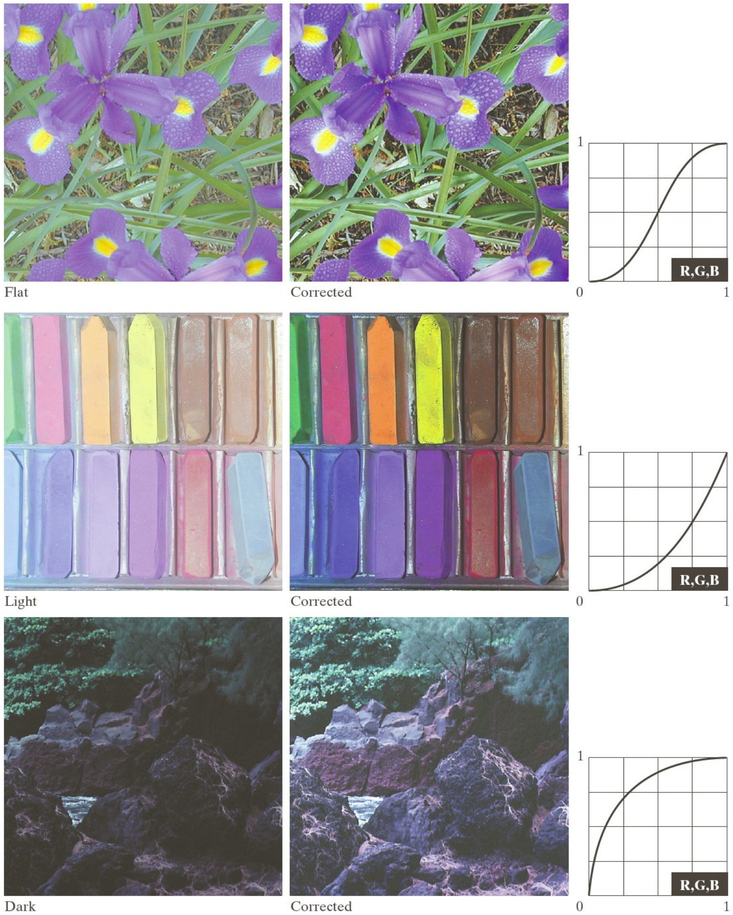


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

Simple Color Transformations

- Tonal correction, gamma correction. Different functions depending on the color

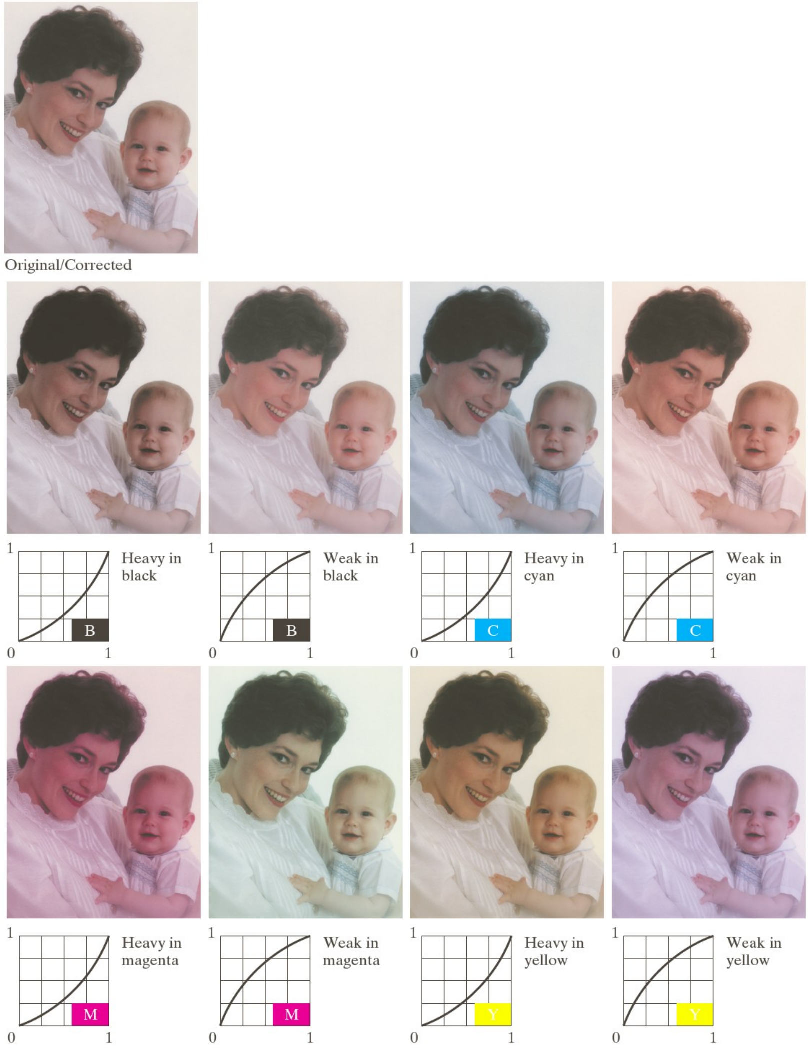
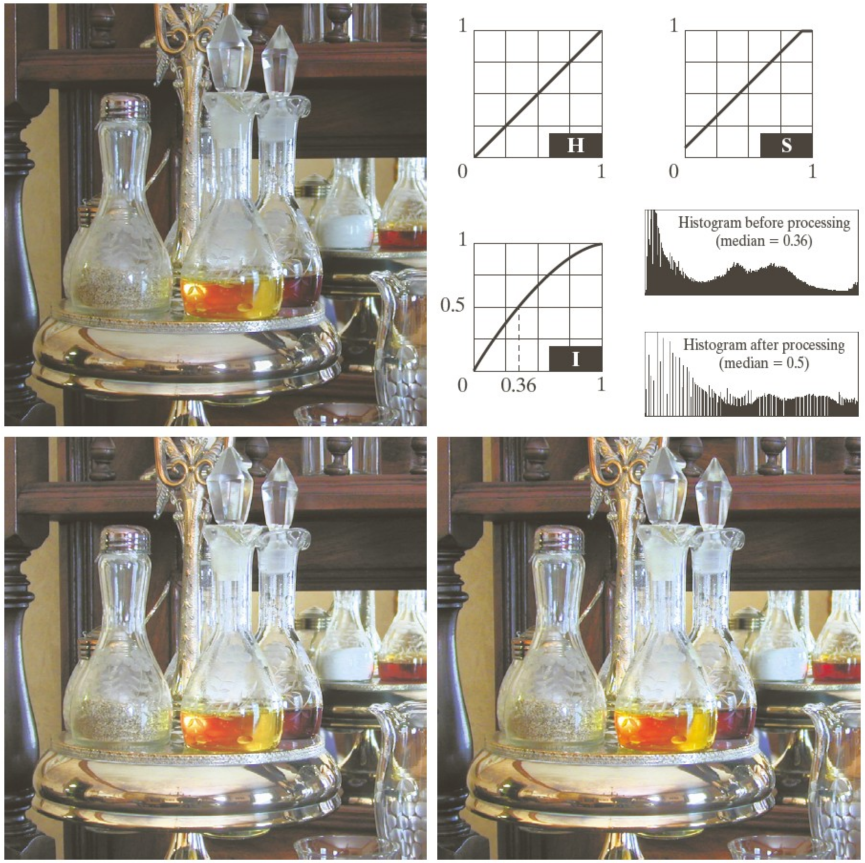


FIGURE 6.36 Color balancing corrections for CMYK color images.

Color Histogram Processing

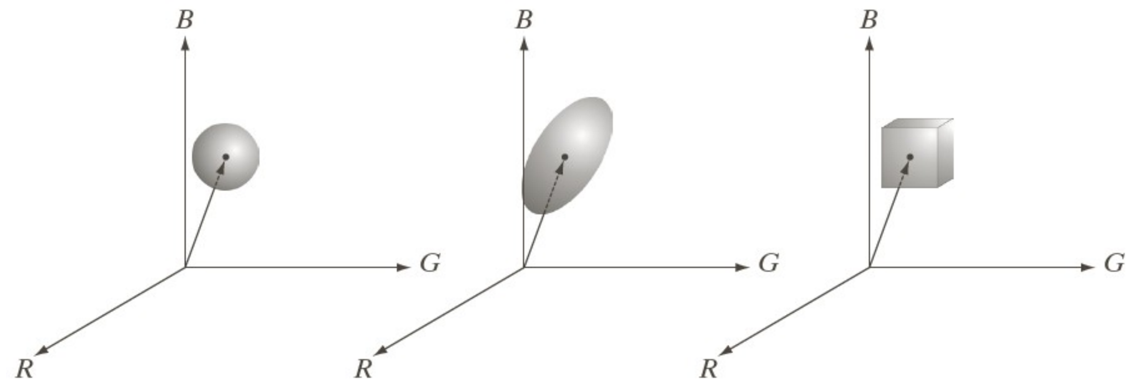


a b
c d

FIGURE 6.37 Histogram equalization (followed by saturation adjustment) in the HSI color space.

On the I channel of HSI
Histogram processing each RGB channel independently will change the colors of the image

Color Slicing

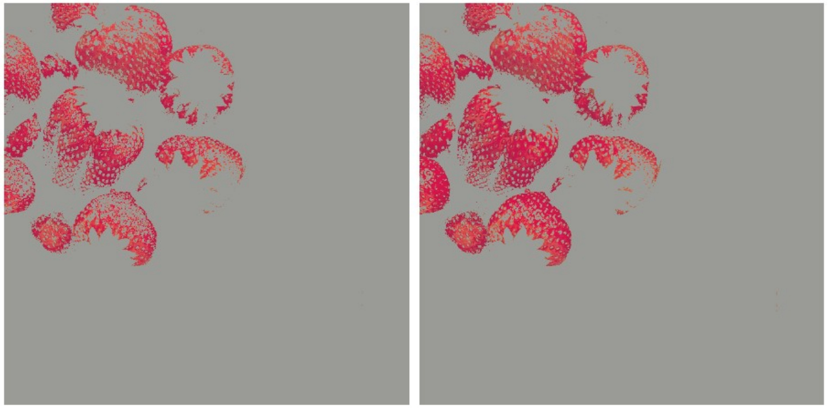


- Slice out a narrow color band from the full color image
- Set all other colors to some backgound color (0.5, 05, 0.5)
- Slice out colors around a prototype color $(a_1,..,a_n)$

$$s_i = \begin{cases} 0.5, & \left[|r_j - a_j| > D \right]_{1 \le j \le n} \\ r_i, & otherwise \end{cases}$$

$$s_i = \begin{cases} 0.5, & \sum_{j=1}^{n} (r_j - a_j)^2 > R^2 \\ r_i, & otherwise \end{cases}$$

The result differs between cube and sphere.
Based on red (0.69,0.16,0.19)

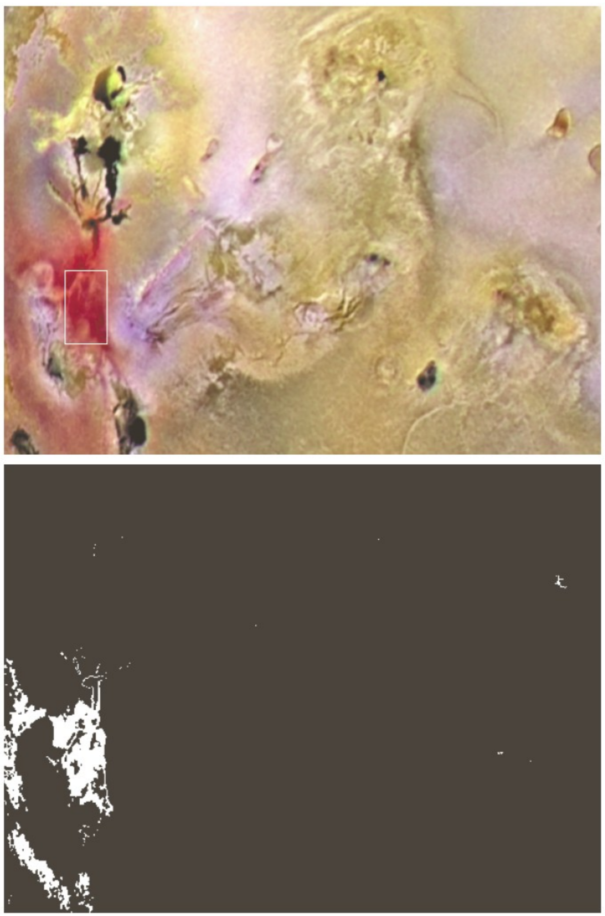


a b

FIGURE 6.34 Color-slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at (0.6863, 0.1608, 0.1922), and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5).

Color segmentation

Similar to color slicing, distance D from a given color **a**



a
b

FIGURE 6.44 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 6.42(h).

$$D(z,a) = \|z-a\|$$

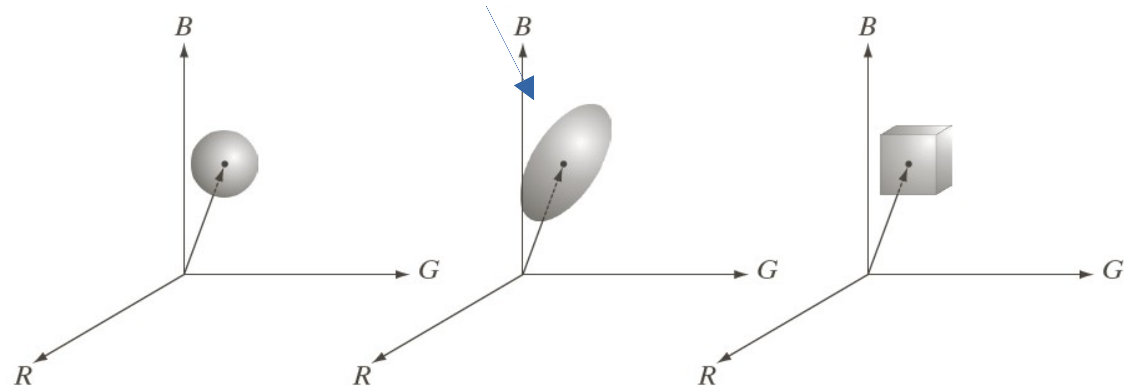
Euclidean norm in color space

$$= [(z-a)^T(z-a)]^{\frac{1}{2}}$$

$$= [(z_R-a_R)^2 + (z_G-a_G)^2 + (z_B-a_B)^2]^{\frac{1}{2}}$$

$$D(z,a) = [(z-a)^T C^{-1}(z-a)]^{\frac{1}{2}}$$

C 3x3 covariance matrix



Smoothing and Sharpening

- Smoothing and sharpening RGB images can be performed componentwise
- An alternative is to smooth or sharpen only the I component in HSI
- Example, smoothing:

$$\bar{c}(x,y)=\frac{1}{K}\sum_{(s,t)\in S_{xy}} c(s,t)=\begin{bmatrix} \frac{1}{K}\sum_{(s,t)\in S_{xy}} R(s,t) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} G(s,t) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} B(s,t) \end{bmatrix}$$

Filter for each channel

Reminder: I= (R+G+B)/3

Smoothing and Sharpening

RGB

a b
c d

FIGURE 6.38 (a) RGB image. (b) Red component image. (c) Green component. (d) Blue component.

HSI

a b c

FIGURE 6.39 HSI components of the RGB color image in Fig. 6.38(a). (a) Hue. (b) Saturation. (c) Intensity.

Result of smoothing

FIGURE 6.40 Image smoothing with a 5×5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

Close but not exactly the same.
Color can be distorded in RGB if the filter has a large size

Result of sharpening

a b c

FIGURE 6.41 Image sharpening with the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the HSI intensity component and converting to RGB. (c) Difference between the two results.

Denoising

- In principle noise affect all colors in the same way.
- If not, RGB denoising may be more efficient
- Noise in one color channel -> noise in all HSI channels

- Conclusion: the best representation is different depending on the task

a b
c d

FIGURE 6.50 (a) RGB image with green plane corrupted by salt-and-pepper noise. (b) Hue component of HSI image. (c) Saturation component. (d) Intensity component.