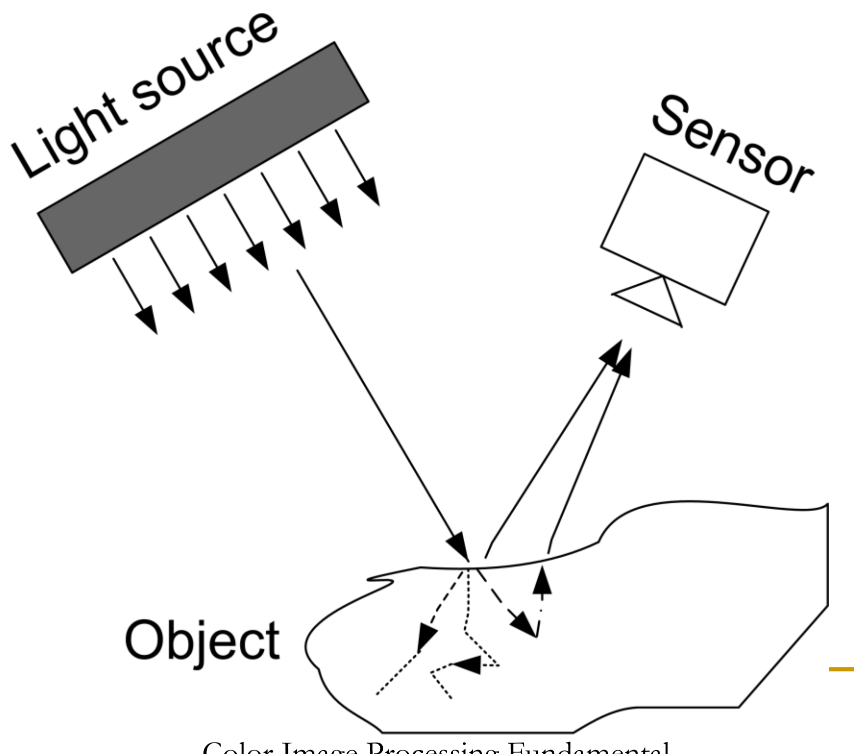
Color image processing: fundamentals

What are Colors?

- The colors that humans and cameras perceive are determined by the nature of the light reflected from an object!
- Green objects reflect “green” light!



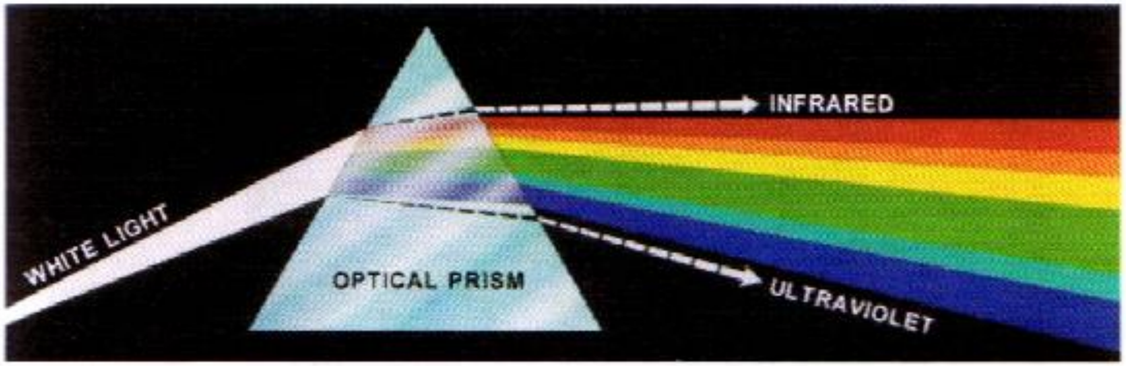
- The perception of color is very important for humans.
- **Humans use color information** to distinguish objects, materials, food, places, and even the time of day.
- Color is especially convenient because it provides multiple measurements at a single pixel of the image, often enabling classification to be done without complex spatial decision-making.

- Color image processing is motivated by two main factors:
  1. Color is a powerful descriptor that often simplifies object identification and extraction from a scene.
  2. We can distinguish between thousands of color shades and intensities, compared to about 20-30 values of gray.

- There are two major areas of color image processing:
  1. Full-color processing – images are acquired with a full-color sensor (TV camera, color scanner);
  2 Pseudocolor processing -- colors are assigned to a particular monochrome intensity or intensity range.
- Many methods discussed for processing of monochrome images are applicable to color images; other methods need reformulation.

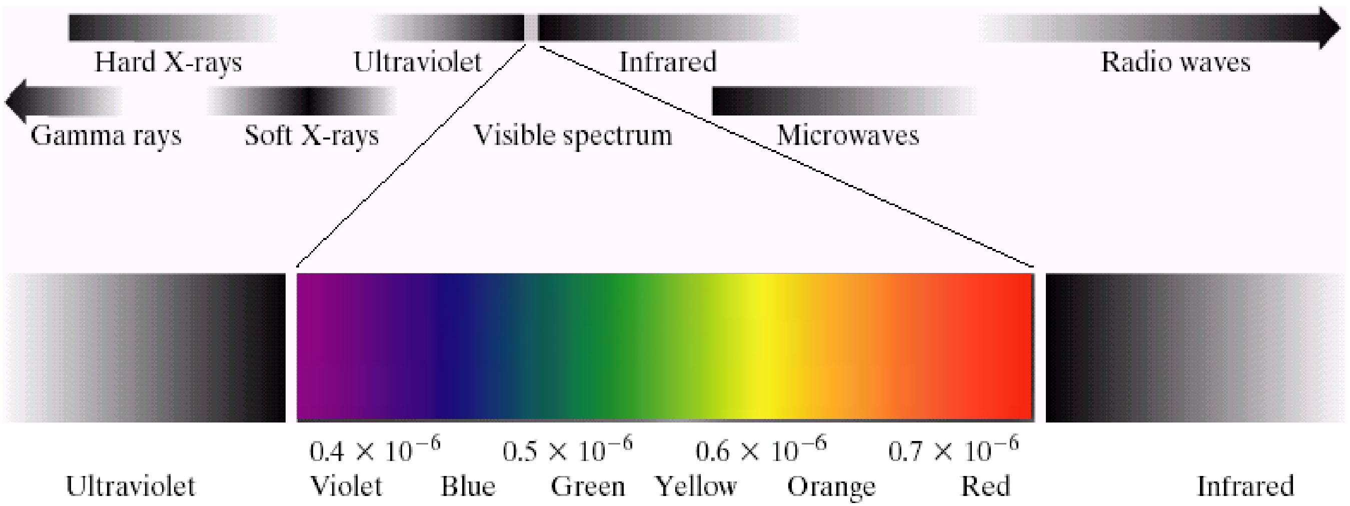
Fundamentals of color

- Mechanisms of color processing by human brain are not completely understood. However, the physical nature of color can be formally expressed and modeled.



1666: Sir Isaac Newton managed to pass a beam of white light through a glass prism and got a rainbow on the other side where each color blends smoothly into the next

- **Chromatic:** Light waves; Visual range: 400nm-700nm



- Particular colors of the object as human perceive them are determined by the nature of light reflection properties of the object.
- A body reflecting light that is balanced in all visible wavelengths appears as white.
- A body reflecting more light in a particular range of wavelengths and absorbing light in other bands appears as colored.

Note: creatures other than humans may (and do!) perceive colors in a completely different way than we do…

- Color sensitive sensors of human retina are cones. The total of 6-7 million of cones in a human eye can be associated with one of three groups:
  - red (~575 nm) sensitive (about 65 % of all cones)
  - green (~535 nm : 33%)
  - blue (~445 nm : 2%) sensitive.

- Due to the absorption characteristics of the human eye, we see colors as variable combinations of so-called ***primary colors of light***:
  - Red (R)
  - Green (G)
  - Blue (B).

- The following wavelengths are designated to them in 1931: 700 nm, 546.1 nm, and 435.8 nm.

Note: **this selection of primary colors is rather arbitrary: about every three colors whose wavelengths are far enough apart can serve as “primary”.**

- Adding primary colors of light produces ***secondary colors***:
  - magenta (red + blue) (品紅)
  - cyan (green + blue) (青色)
  - yellow (green+ red)
- Mixing the three primary colors of light (or a secondary with its opposite primary color) ***in the right intensities*** produces white light. Mixing together the three secondary colors of light, black (no light) can be produced.

- Colors are usually distinguished from each other through the three characteristics:
  - **brightness** （亮度）
  - **hue** （色調）
  - **Saturation** （飽和度）

- Brightness (a subjective descriptor **practically impossible to measure** since it is associated to both the intensity and color sensation)
- Hue represents dominant color as perceived by an observer (red, yellow, blue).
- Saturation is the amount of white added to a hue (purity of the color). For example, we need to specify saturation to characterize pink (red+ white)

Color Models

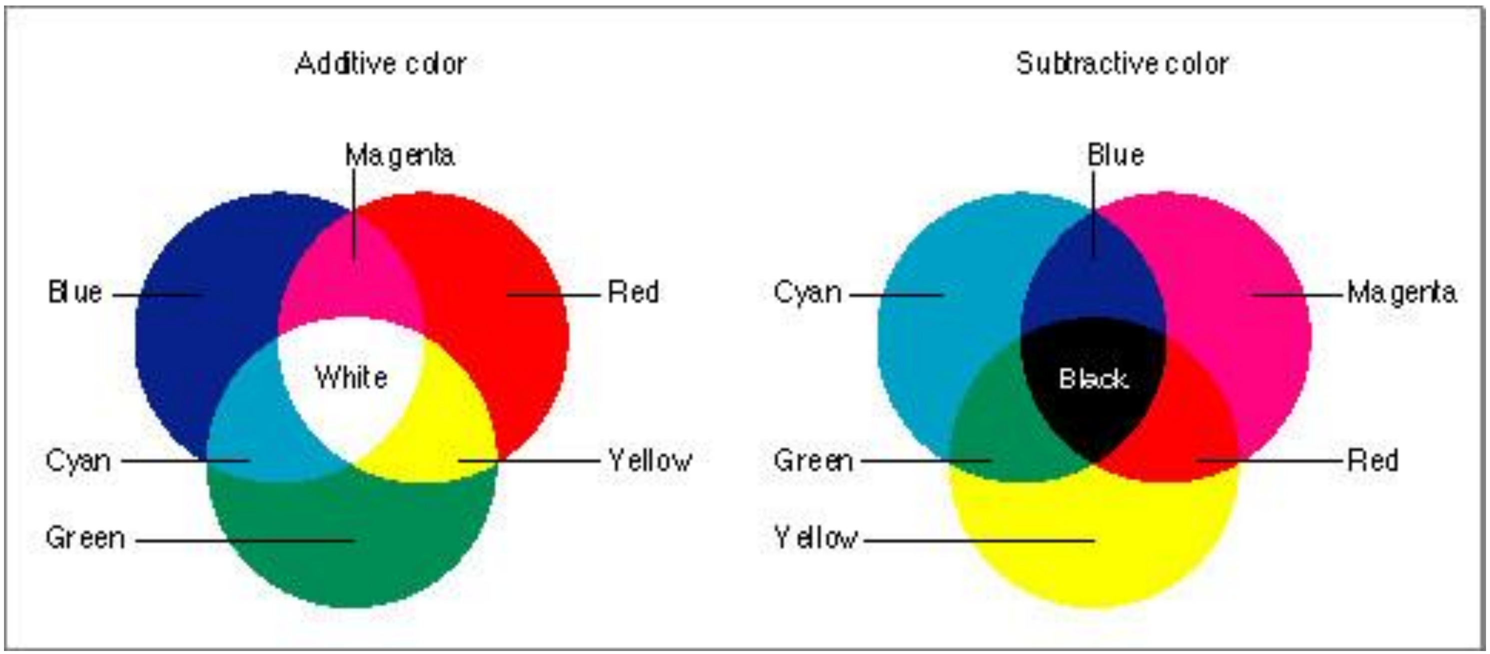
- The purpose of a color model (also called color space or color system) is to facilitate the specification of colors in some standard, generally accept way.
- **RGB** (red,green,blue) : monitor, video camera.
- **CMY**(cyan,magenta,yellow),**CMYK** (CMY, black) model for color printing.
- **HSI** model, which corresponds closely with the way humans describe and interpret color.

Red, Green, Blue

- R,G,B are called **Primary Colors**
- R,G,B where chosen due to the structure of the human eye
- R,G,B are used in cameras

R+G+B=White? Really?!?

- So why don't we get white, when we use paint? *Subtractive Color!*
- But why does it work for the TV? *Additive Color!*

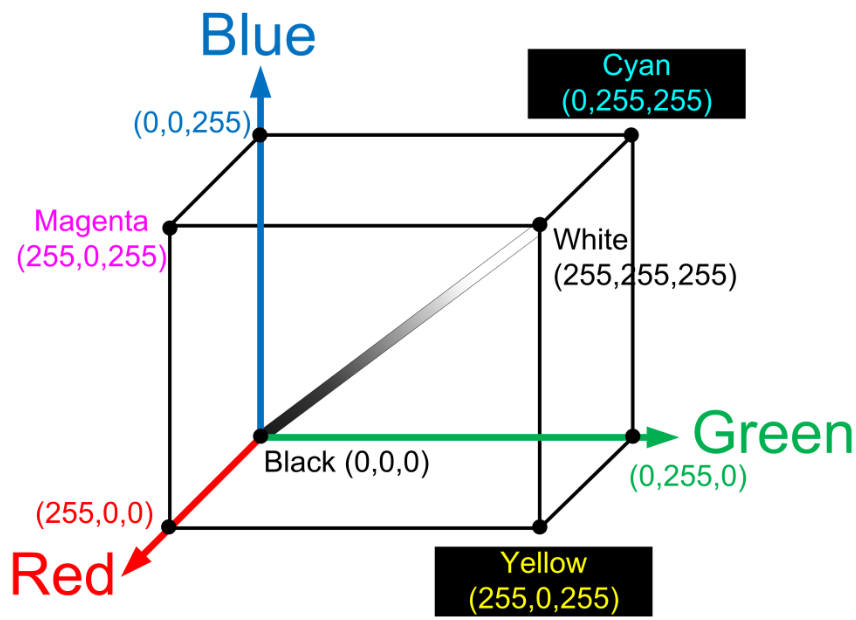


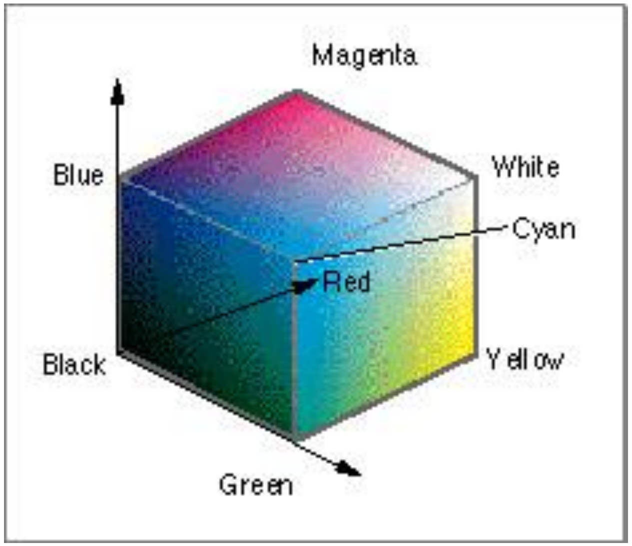
Additive/Subtractive Color

- **Additive Color**: Sum of light of different wave lengths. That light reaches our eye directly.
  - Examples: TV, Multimedia Projector
- **Subtractive Color**: White Color is emitted by the sun and is only partly reflected from an object!
  - Red paint filters all light, except red!
  - Yellow paint absorbs blue, but reflects red and green

RGB Color Space

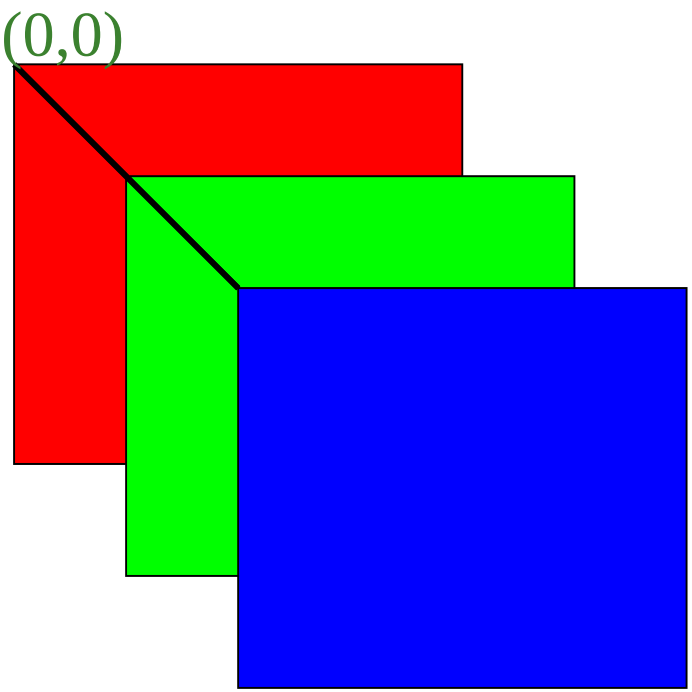
- The “classical” Computer Color space
- 3 different colors: Red, Green, Blue
- If R,G,B have the same energy, we perceive a shade of grey





RGB Color Space

RGB Color Space



A single pixel consists of three components: [0,255].
Each pixel is a **Vector**.

128	251	60

=

Pixel-Vector in the computer memory

Final pixel in the image

Caution! Sometimes pixels are not stored as vectors. Instead, first is stored the complete red component, then the complete green, then blue.

Example RGB

Original Image

R-Component

G-Component

B-Component

Convert color to grayscale

?

1. Average method: I = (R+G+B) / 3

(

+

+

) / 3

2. **Luminosity** method: a weighted average to account for human perception, We're more sensitive to green than other colors, so green is weighted most heavily.

$I = a1*R + a2*G + a3*B \qquad a1+a2+a3=1$

The formula for luminosity is

$I = 0.21\,R + 0.71\,G + 0.08\,B.$

3. The **lightness** method averages the most prominent and least prominent colors:

   $I = (\max(R, G, B) + \min(R, G, B)) / 2$

- The lightness method tends to reduce contrast
- The luminosity method works best overall

Original Image

Average Image

Luminosity Image

Lightness Image

Color thresholding

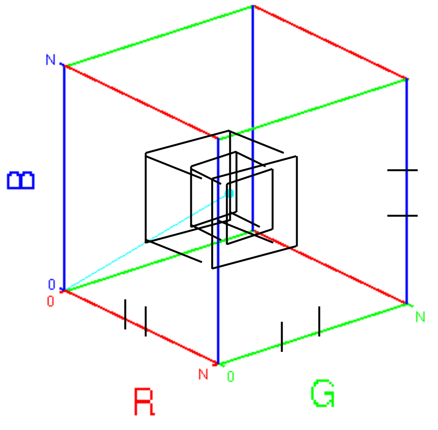
- For example: we want to detect red objects using the color
- 3 thresholding in RGB
- If: Tredmin < R < Tredmax AND

  Tgreenmin < G < Tgreenmax AND

  Tbluemin < B < Tbluemax

  Then: object pixel

  Else: non-object pixel
- Problem:
  - If the intensity changes in different color,
    we need larger T
    => non-object pixels will be detected

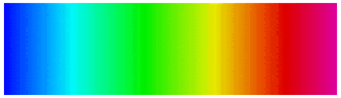


- RGB thresholding reflects the way the camera works and how the data is stored in the computer, but it does not correspond to the way that people recognize color. Therefore, the HSI color models are more often used.

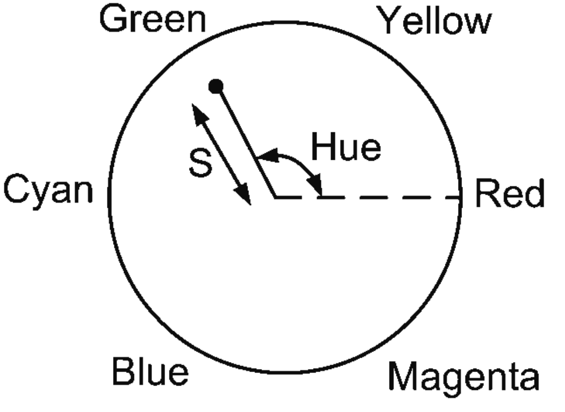
HSI

- H=Hue   S=Saturation I=intensity

- H and S may characterize a color: **Chromaticities**
- **Hue**: The dominant wavelength in the mixture of light waves, as perceived by an observer.
  - Hue is the color attribute that describes a pure color

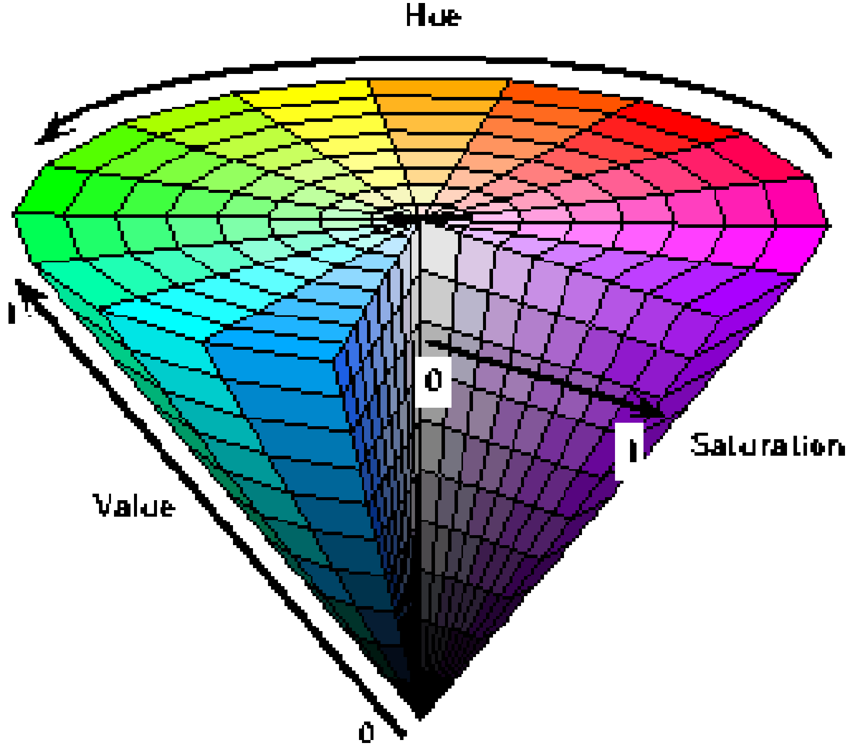


- **Saturation**: relative purity; inverse of the amount of white light mixed with hue 
  - **Example:** Pure colors are fully saturated. Not saturated are for example pink (red+white)

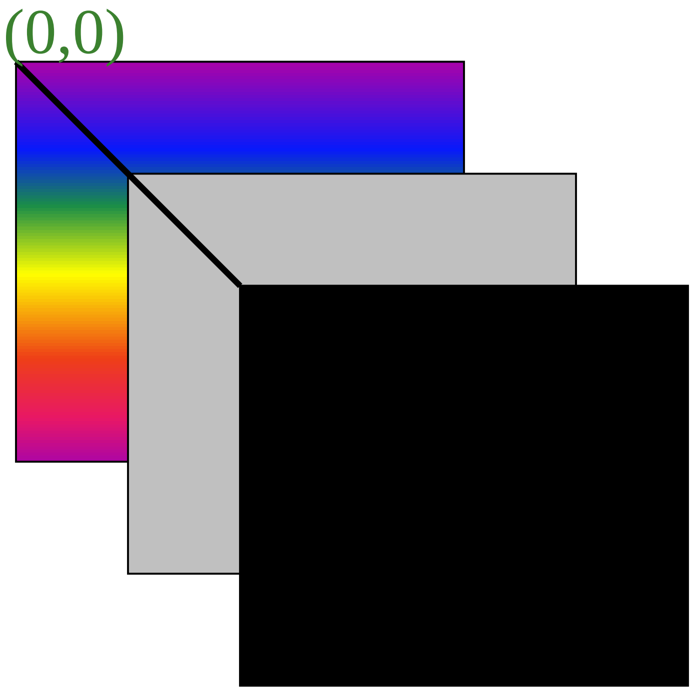


HSI color space

• Perhaps the most intuitive color representation!



HSI Color Space



A single pixel consists of three components.

Each pixel is a **Vector**

128	251	60

=

Pixel-Vector in the computer memory

Final pixel in the image

Caution! Sometimes pixels are not stored as vectors. Instead, first is stored the complete hue component, then the complete sat., then the intensity.

Example HSI

Original Image

Hue

Saturation

Intensity

RGB→HSI

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

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

$$H = \left(\begin{cases} \theta & G \geq B \\ 2\pi - \theta & G < B \end{cases}\right) / (2\pi)$$

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

$$If\ S = 0, H = 0$$

HSI→RGB

$$(1)\ H \in [0, 2\pi/3]$$

$$B = I(1 - S)$$

$$R = I\left[1 + \frac{s\cos H}{\cos(\frac{\pi}{3} - H)}\right]$$

$$G = 3I - (B + R)$$

$$(2)\ H \in [2\pi/3, 4\pi/3]$$

$$R = I(1 - S)$$

$$G = I[1 + \frac{s\cos(H - \frac{2\pi}{3})}{\cos(\pi - H)}]$$

$$B = 3I - (R + G)$$

$$(3)\ H \in [4\pi/3, 2\pi]$$

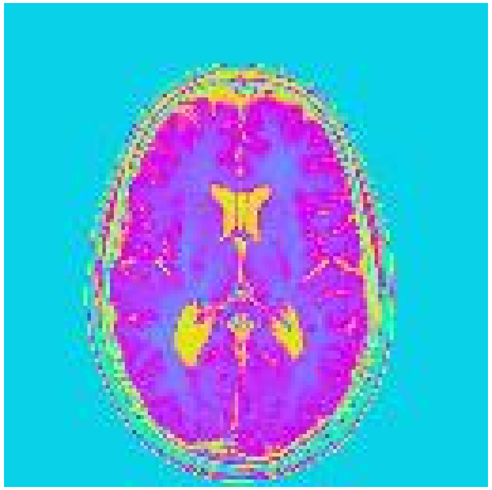
$$G = I(1-S)$$

$$B = I[1 + \frac{s\cos(H - \frac{4}{3}\pi)}{\cos(\frac{5\pi}{3} - H)}]$$

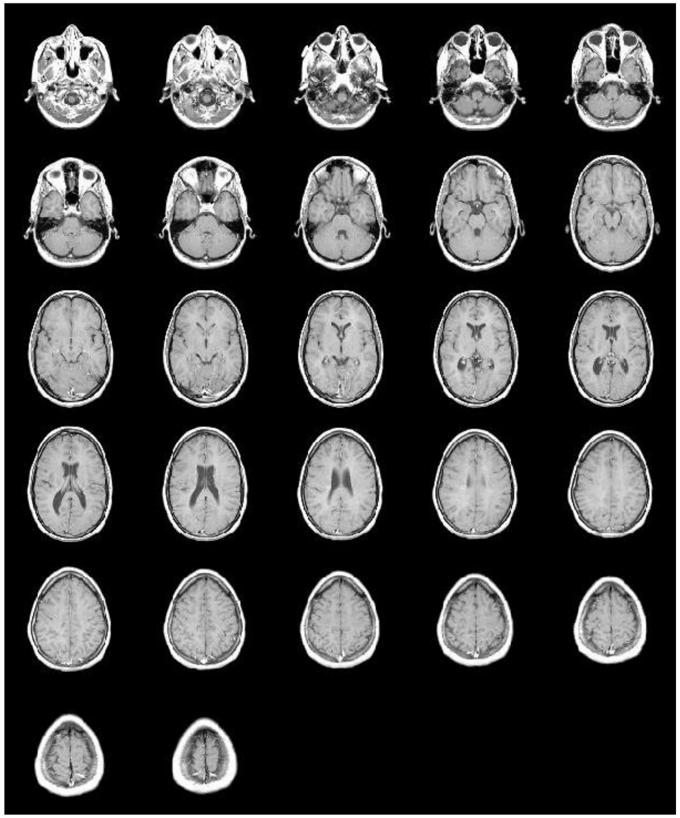
$$R = 3I - (G+B)$$

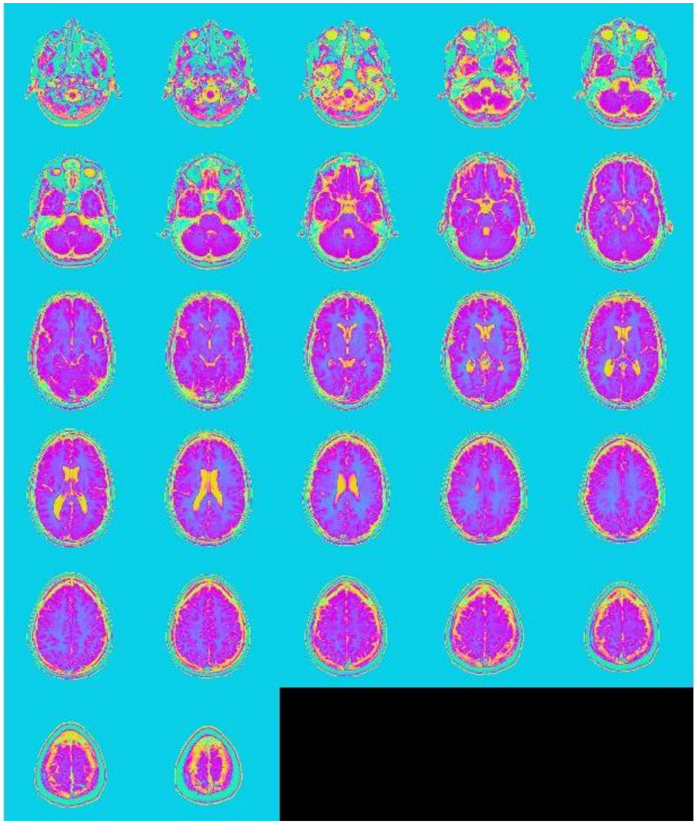
Full Color / Pseudo Color

- **Full Color**: acquired by a TV camera/scanner
- **Pseudo Color**: Assigning a shade of color to a monochrome intensity or range of intensities

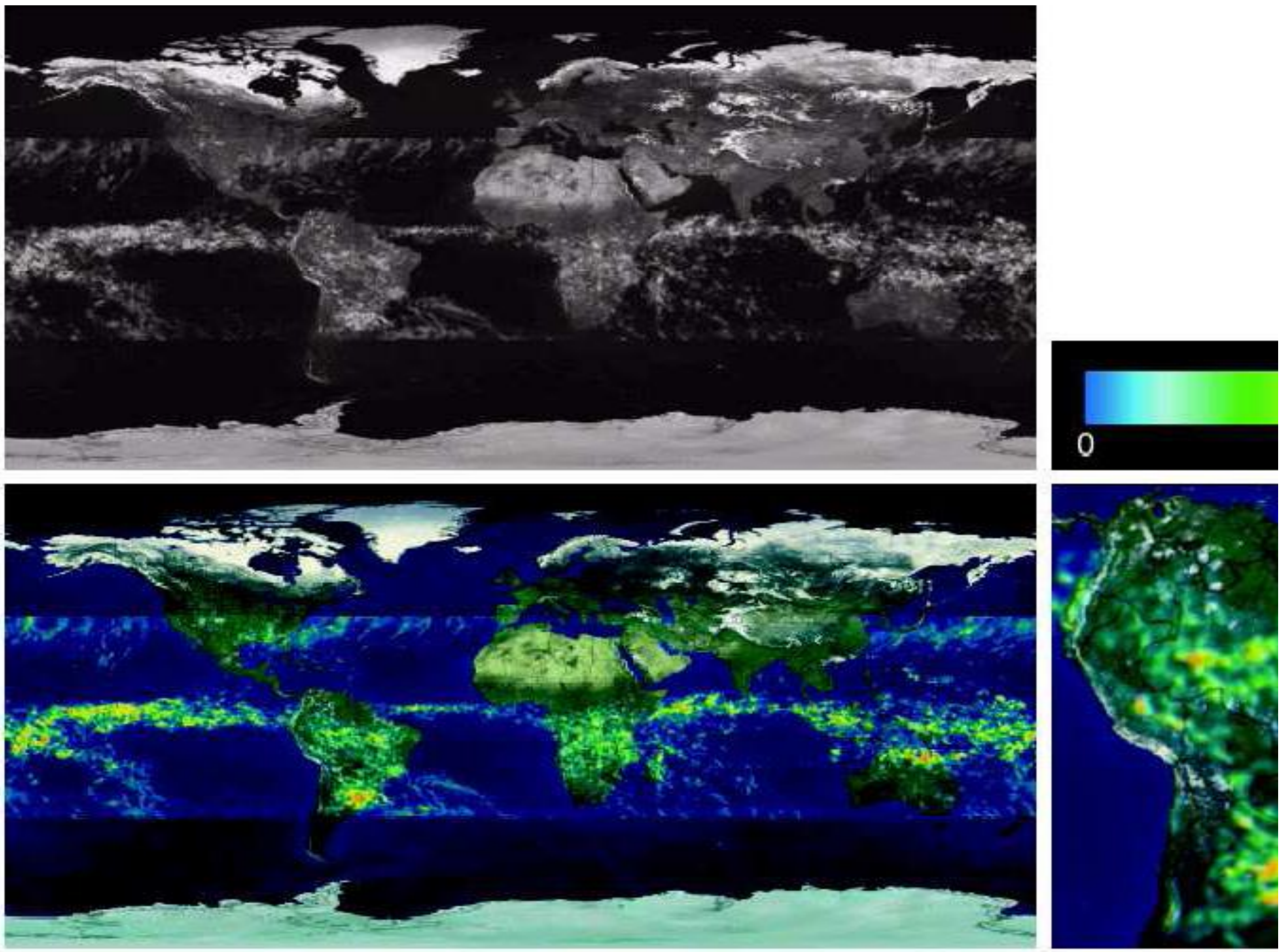


Mapping of Gray Values into Pseudo Color Images





Pseudo Color (Remote Sensing Image遙感圖像)



a b
c d

Different usage of RGB and HSI color models.

- HSI Models is especially useful for image processing on the isolated intensity component (preserves color).
- RGB is typically used for computer graphics.

Color Image Histogram Processing

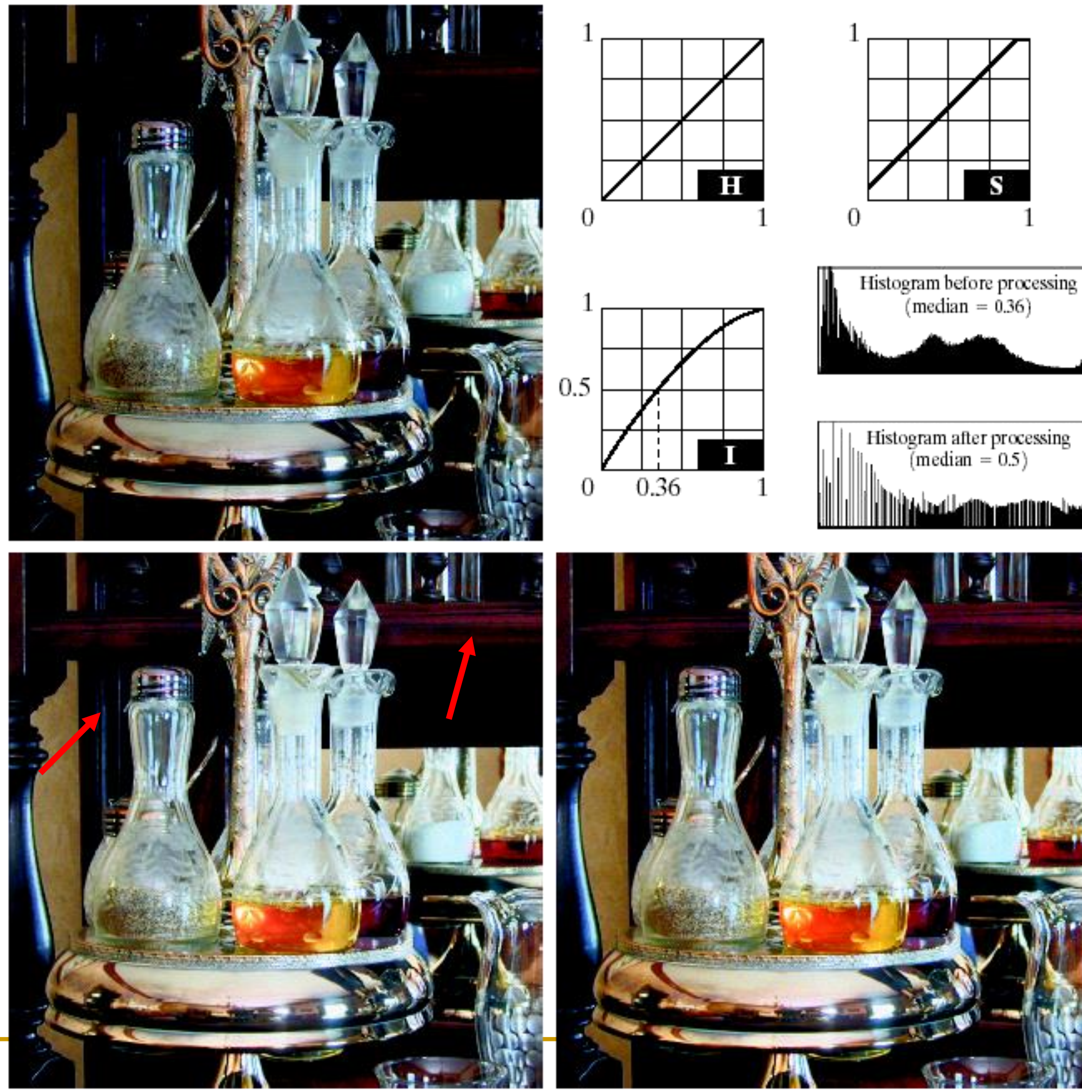
- To improve image contrast without ruining the color content, the histogram equalization must be performed on the **intensity component** only.
- Images are converted to HSI, isolate the intensity channel, perform histogram equalization, and convert back to RGB.

Histogram equalization Algorithm

- convert from RGB to HSI model
- get intensity
- Histogram equalization
- convert from HSI to RGB model
- Output RGB file

Full-Color Image Processing

Color Transformation: Histogram Processing



a b
c d

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

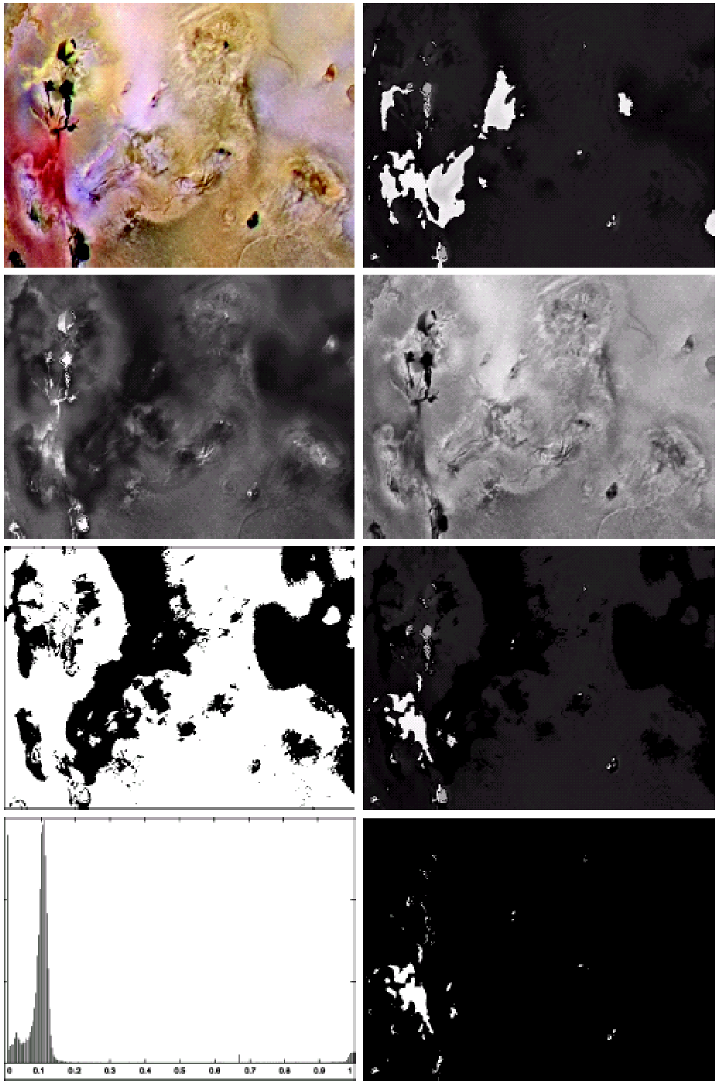
Saturation Adjustment

Segmentation in HSI space

- S: as a template, determine the value of the threshold:
  - $S(i,j)=1$, if $S(i,j)>threshold$
  - $S(i,j)=0$, if $S(i,j)<threshold$
- H: do the thresholding in H: S*H
- I: there is no any color information.

Full-Color Image Processing

Color Segmentation: in HSI Color Space



a b
c d
e f
g h

FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).

Segmentation in RGB space

- Find the color we would like to segment: a
- To every pixel in image (the color difference is small)
- If

$$D(z,a) = \| z - a \| = [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{\frac{1}{2}} \leq D_0$$

- Then I(R,G,B)=1, otherwise I(R,G,B)=0

Color Edge Detection algorithm1

1. Edge detection in R, G, B images using Sobel detector, respectively
2. Combine the three edge images together to get the final edge image

Color Edge Detection algorithm 2

- Suppose c is one of the pixels in RGB image.

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

$$u = \frac{\partial R}{\partial x} r + \frac{\partial G}{\partial x} g + \frac{\partial B}{\partial x} b$$

$$v = \frac{\partial R}{\partial y} r + \frac{\partial G}{\partial y} g + \frac{\partial B}{\partial y} b$$

$$g_{xx} = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2$$

$$g_{yy} = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2$$

$$g_{xy} = \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y}$$

- The direction of the maximum rate of change of c(x,y) is given by the angle

$$\theta = \frac{1}{2}\arctan\left[\frac{2g_{xy}}{(g_{xx} - g_{yy})}\right]$$

- The value of the rate of change in the direction given

$$F(\theta) = \left\{\frac{1}{2}[(g_{xx} + g_{yy}) + (g_{xx} - g_{yy})\cos 2\theta + 2g_{xy}\sin 2\theta]\right\}^{\frac{1}{2}}$$

Full-Color Image Processing

Color Segmentation: Color Edge Detection

a b
c d

FIGURE 6.46
(a) RGB image.
(b) Gradient computed in RGB color vector space.
(c) Gradients computed on a per-image basis and then added.
(d) Difference between (b) and (c).

Full-Color Image Processing

Color Segmentation: Color Edge Detection

a b c

FIGURE 6.47 Component gradient images of the color image in Fig. 6.46. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 6.46(c).

Full-Color Image Processing

Noise in Color Images

a b
c d

FIGURE 6.48 (a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

Full-Color Image Processing

Noise in Color Images

a b c

FIGURE 6.49 HSI components of the noisy color image in Fig. 6.48(d). (a) Hue. (b) Saturation. (c) Intensity.

Full-Color Image Processing

Noise in Color Images

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.

Full-Color Image Processing
Color Image Compression

a b
c d

FIGURE 6.51 Color image compression. (a) Original RGB image. (b) Result of compressing and decompressing the image in (a).

Full-Color Image Processing

Color Transformation: Tone(色調）and Color Correction

The tonal range of an image, also called its key-type, refers to its general distribution of color intensities.

- **High-key images: Most of the information is concentrated at high intensities.**
- **Low-key images: Most of the information is concentrated at low intensities.**

Full-Color Image Processing

Color Transformation: Tonal Correction

Middle-key Image

Flat

Corrected

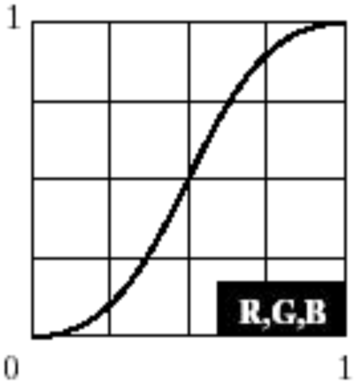


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

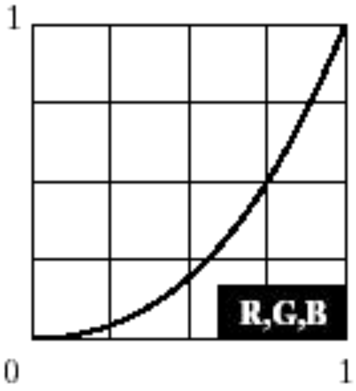
Full-Color Image Processing

Color Transformation: Tonal Correction

High-key Image

Light

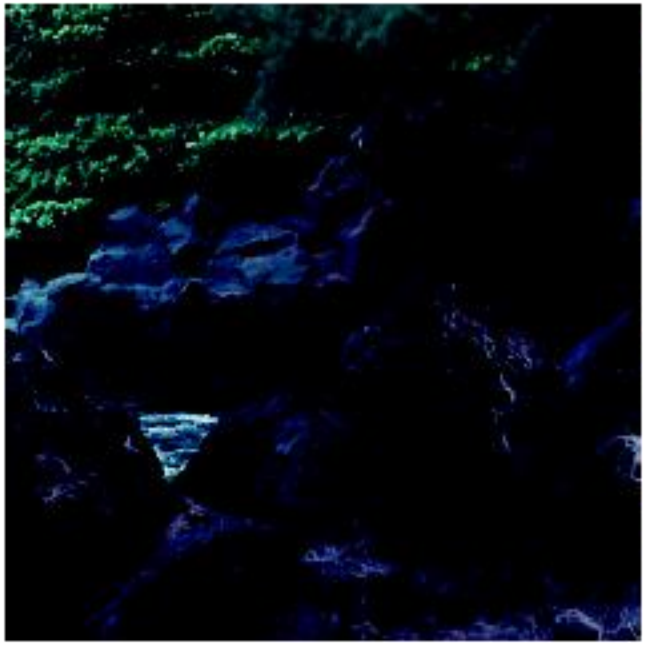
Corrected



Full-Color Image Processing

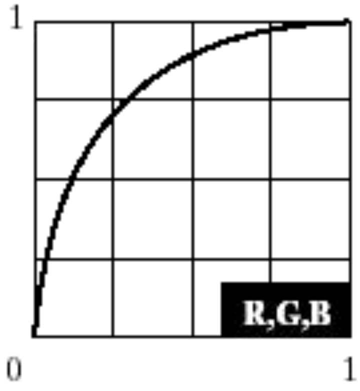
Color Transformation: Tonal Correction

Low-key Image



Dark

Corrected



Full-Color Image Processing

Color Transformation: Color Correction

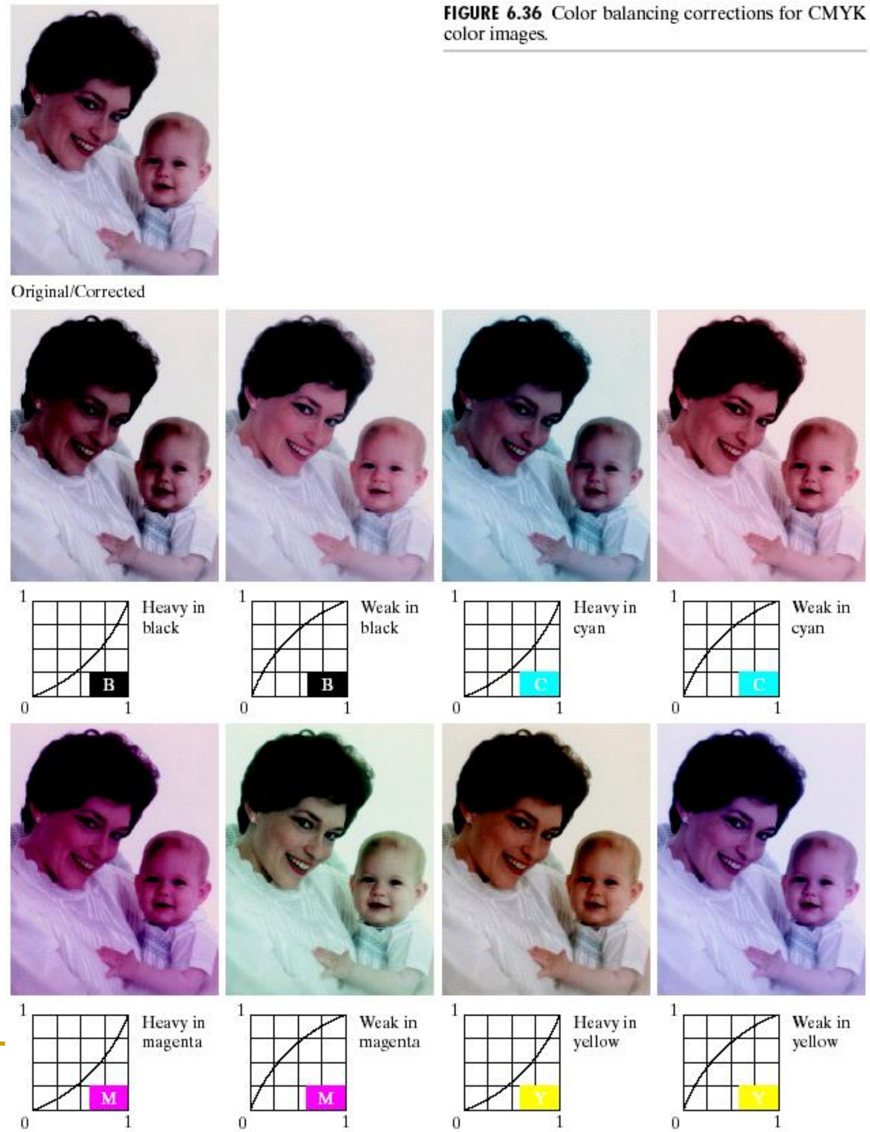


FIGURE 6.36 Color balancing corrections for CMYK color images.

Full-Color Image Processing

Color Image Smoothing

Averaging:

$$\bar{\mathrm{c}}(x, y) = \frac{1}{K} \sum_{(x,y)\in S_{xy}} \mathrm{c}(x, y)$$

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

Full-Color Image Processing

Color Image Smoothing

a b
c d

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

Red

Green

Blue

Full-Color Image Processing

Color Image Smoothing

Hue Saturation Intensity

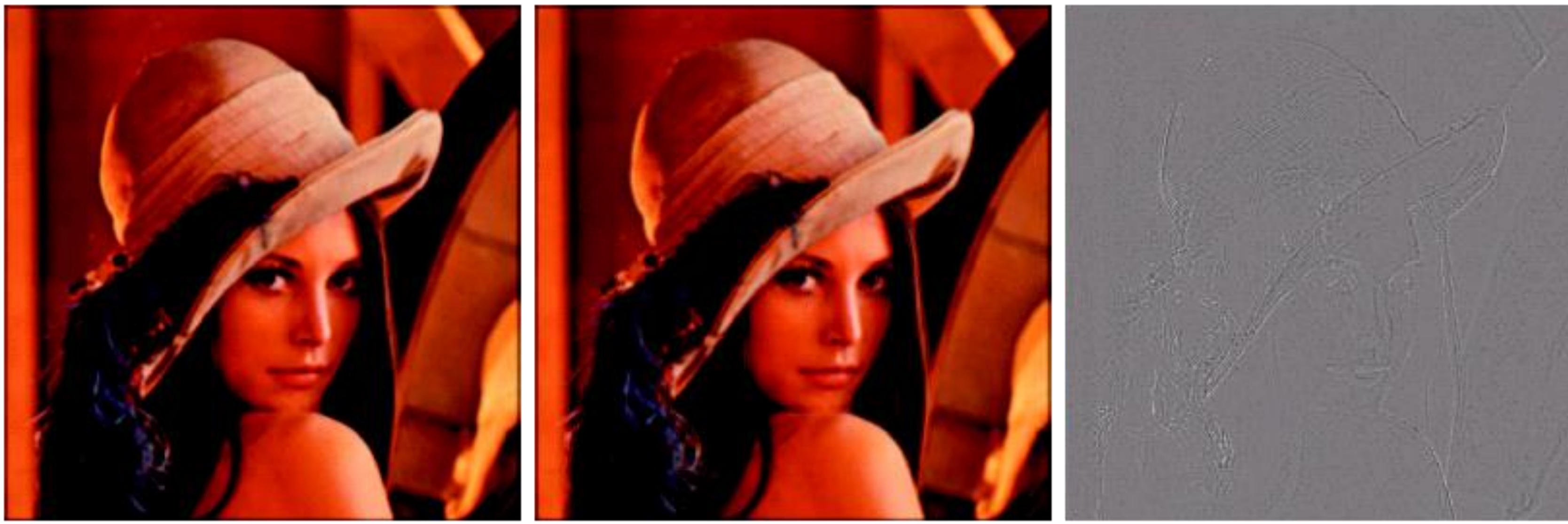
a b c

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

Full-Color Image Processing

Color Image Smoothing

Averaging R,G and B | Averaging Intensity | Difference



a b c

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.

Full-Color Image Processing
Color Image Sharpening

The Laplacian of Vector **c** :

$$\nabla^2\left[\mathbf{c}(x,y)\right]=\begin{bmatrix}\nabla^2 R(x,y)\\ \nabla^2 G(x,y)\\ \nabla^2 B(x,y)\end{bmatrix}$$

Full-Color Image Processing

Color Image Sharpening

Sharpening R,G and B　　Sharpening Intensity　　Difference

a b c

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