Pseudocolor IP

Intensity slicing

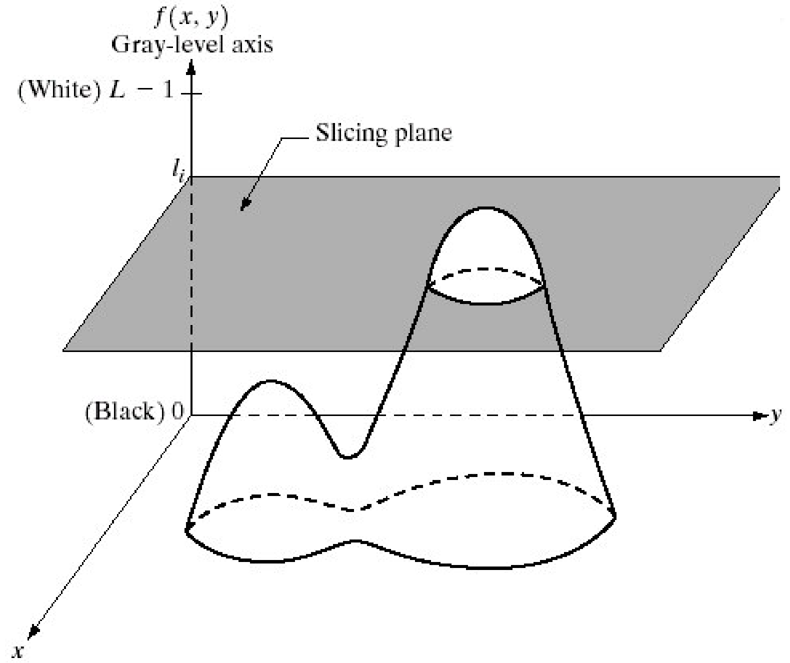


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

if gray > l_i
color A

if gray < l_i
color B.

pseudo color (false color) – assign colors to gray values using some criterion
– used a lot in data visualization

false since these are not real color

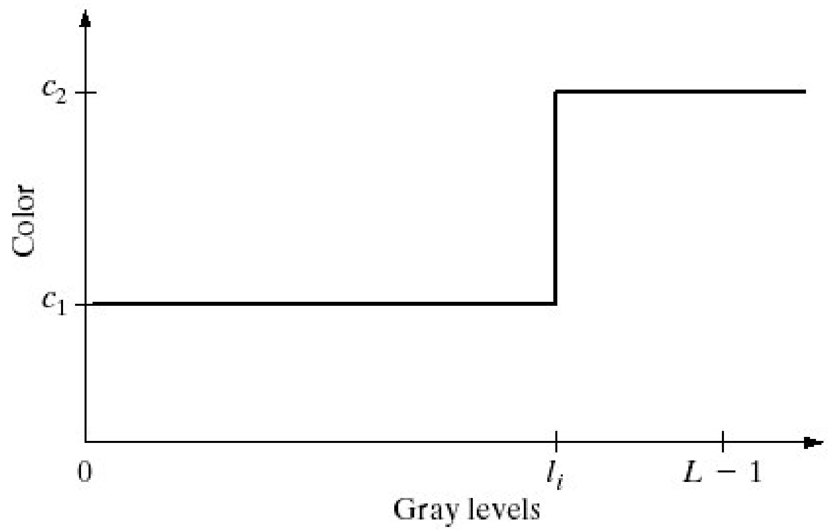
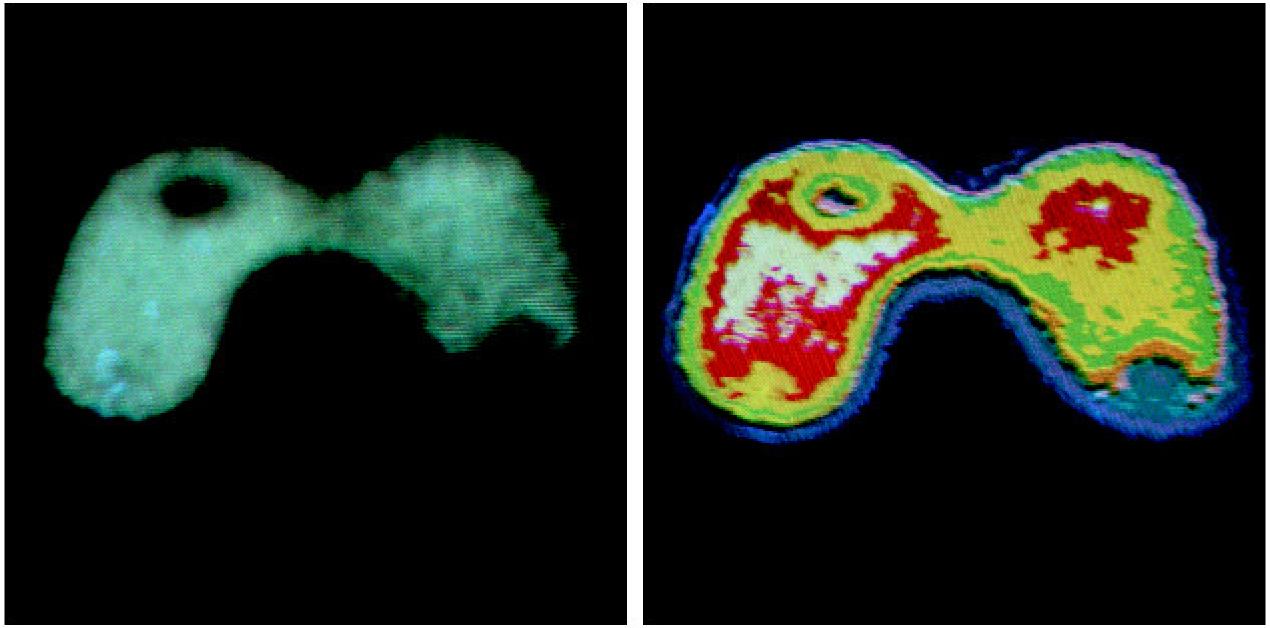


FIGURE 6.19 An alternative representation of the intensity-slicing technique.

Simply another way of describing what is done in Figure 6-18.



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.)

intensity slicing with multiple color slices

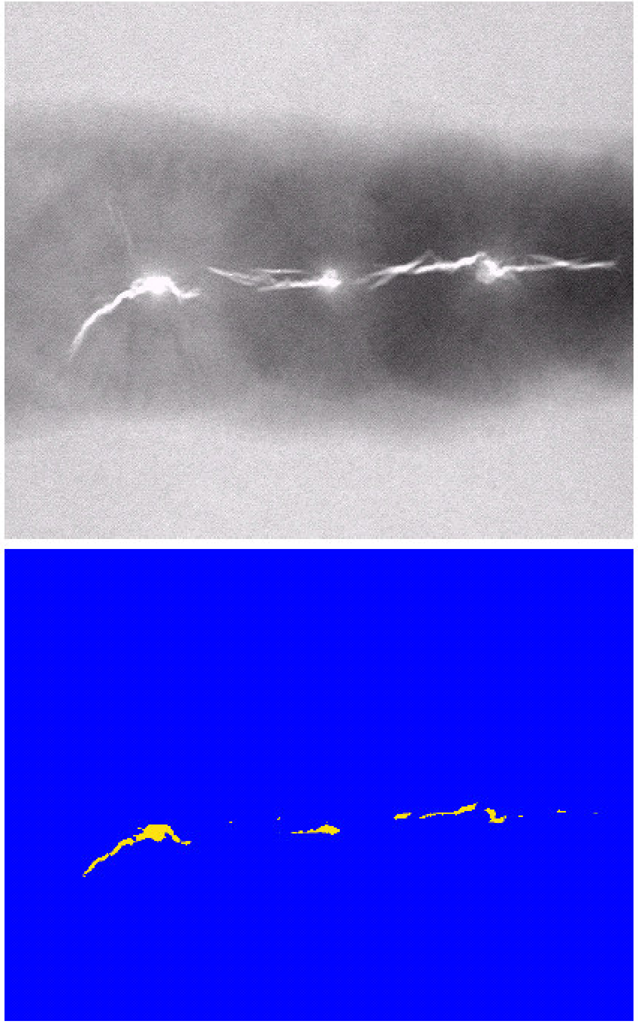
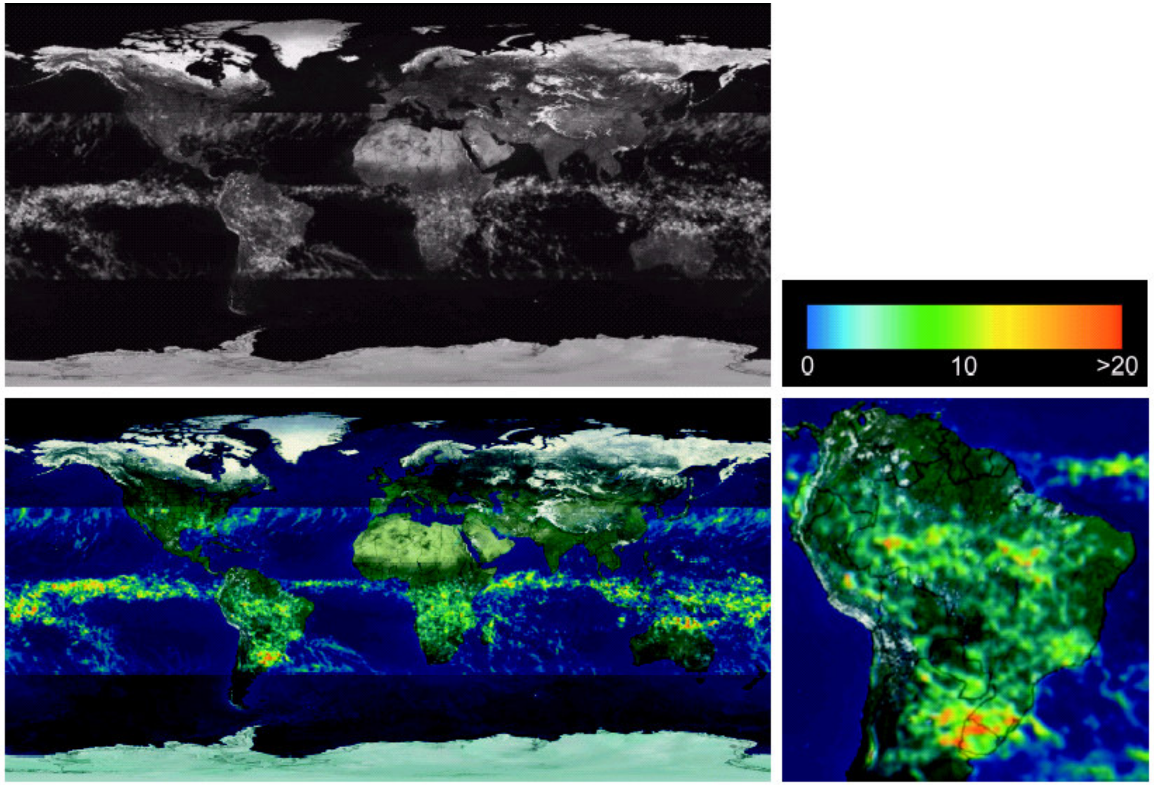


FIGURE 6.21 (a) Monochrome X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)

Simple application in X-ray analysis.
Cracks allow full X-ray intensity through metal.
Image simply codes 255 as yellow and all others as blue for inspection.



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 America region. (Courtesy of NASA.)

Combine signals from (Tropical Rainfall Measuring Mission satellite)

- precipitation radar
- microwave imager
- visible/IR scanner.

to estimate average monthly rainfall.

Difficult to see patterns in grayscale. Much easier to see in pseudocolor.

Gray level to color transform

single gray scale input image

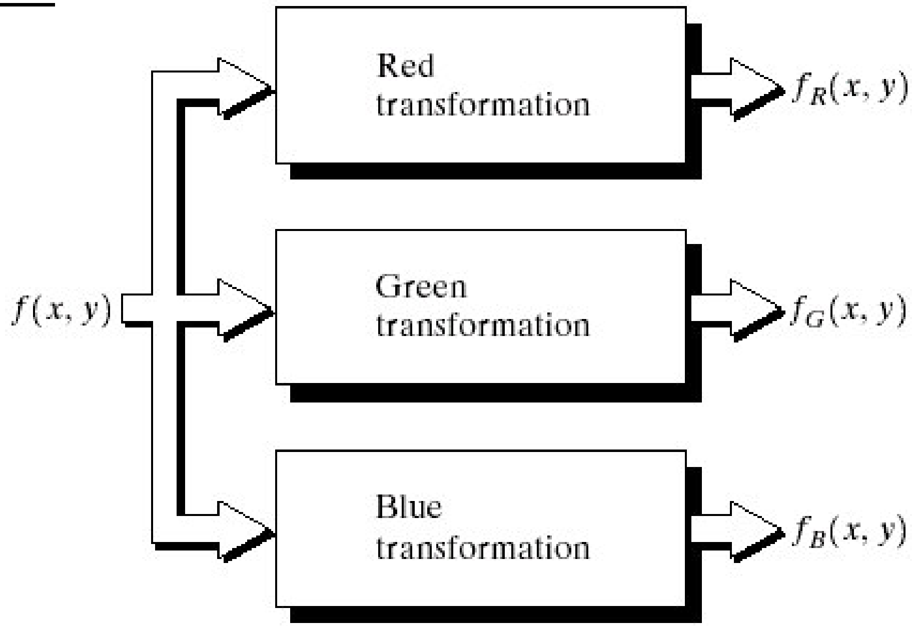
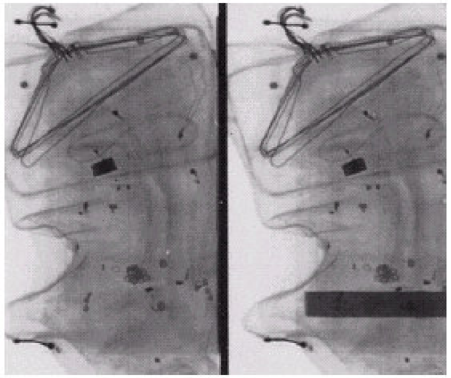
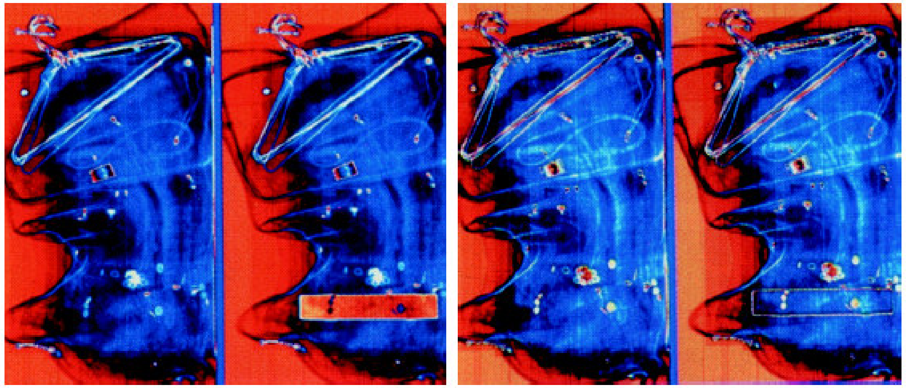


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

We can use simultaneous non-linear transforms to drive a color camera.



2 images, both grey scale

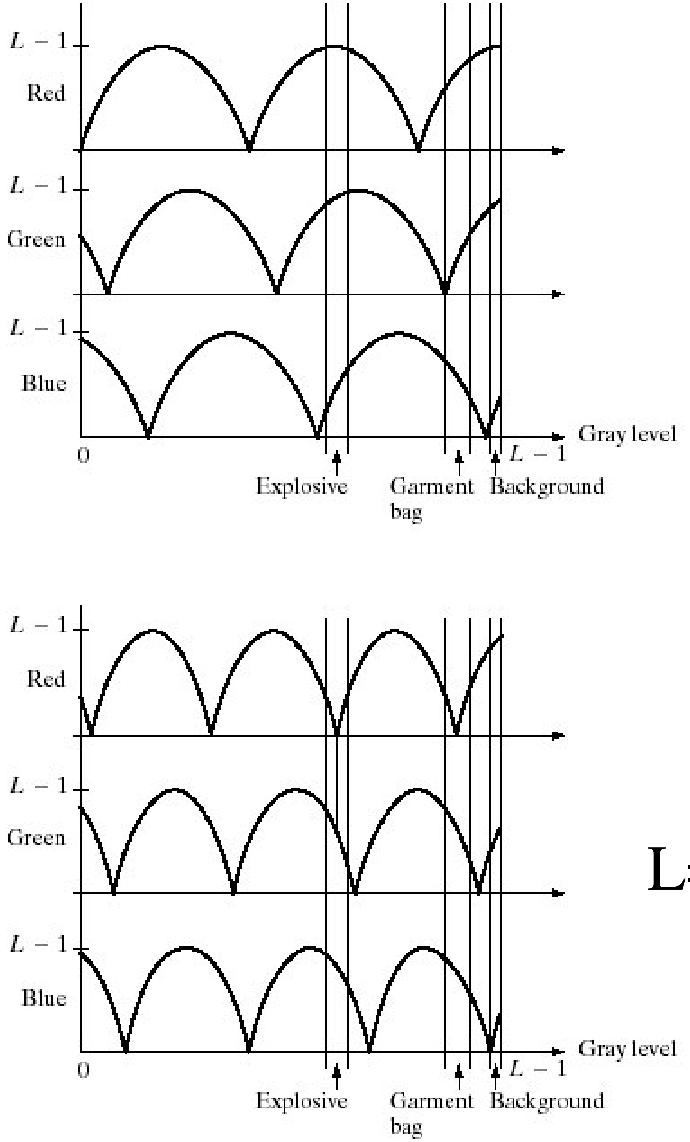


Notice how explosives show up differently

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.)

$$\underbrace{\text{transform } 6.25(a) \qquad \text{transform } 6.25(b)}_{\text{2 different transforms.}}$$



a
b

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

L= number of grey levels

transformations used.
to make explosives in 6.24 visible.

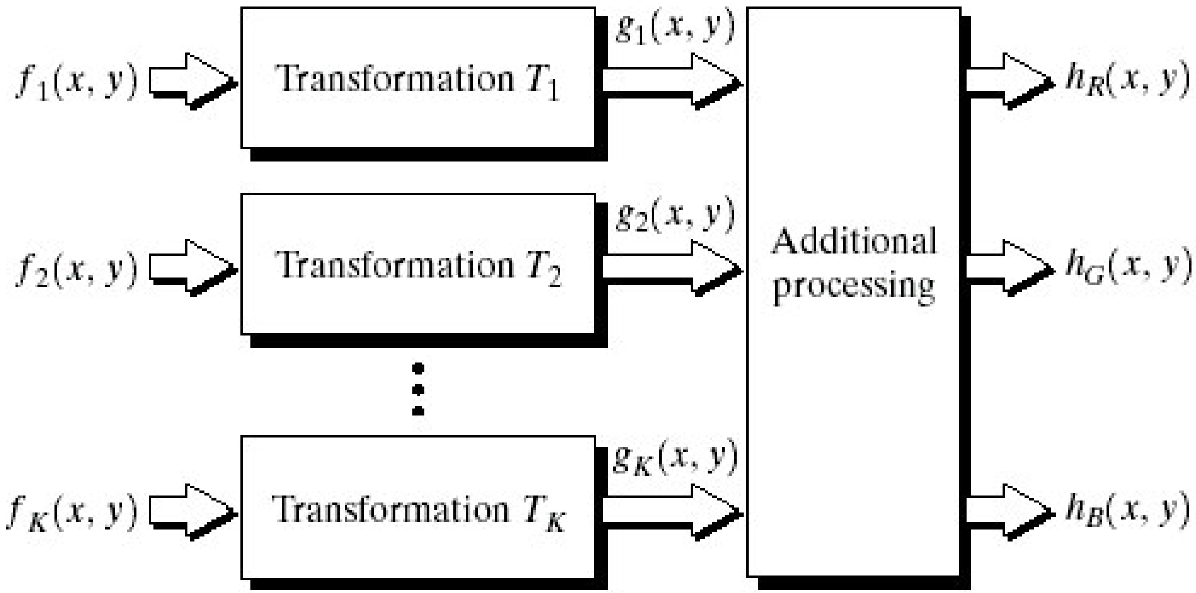
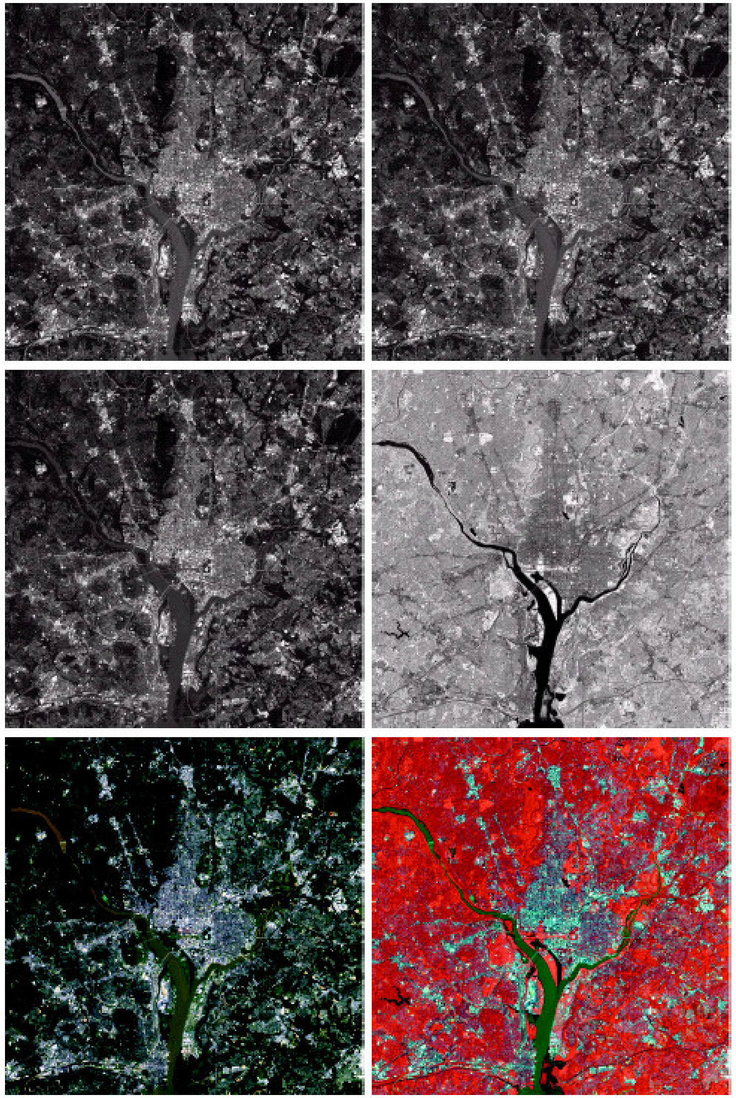


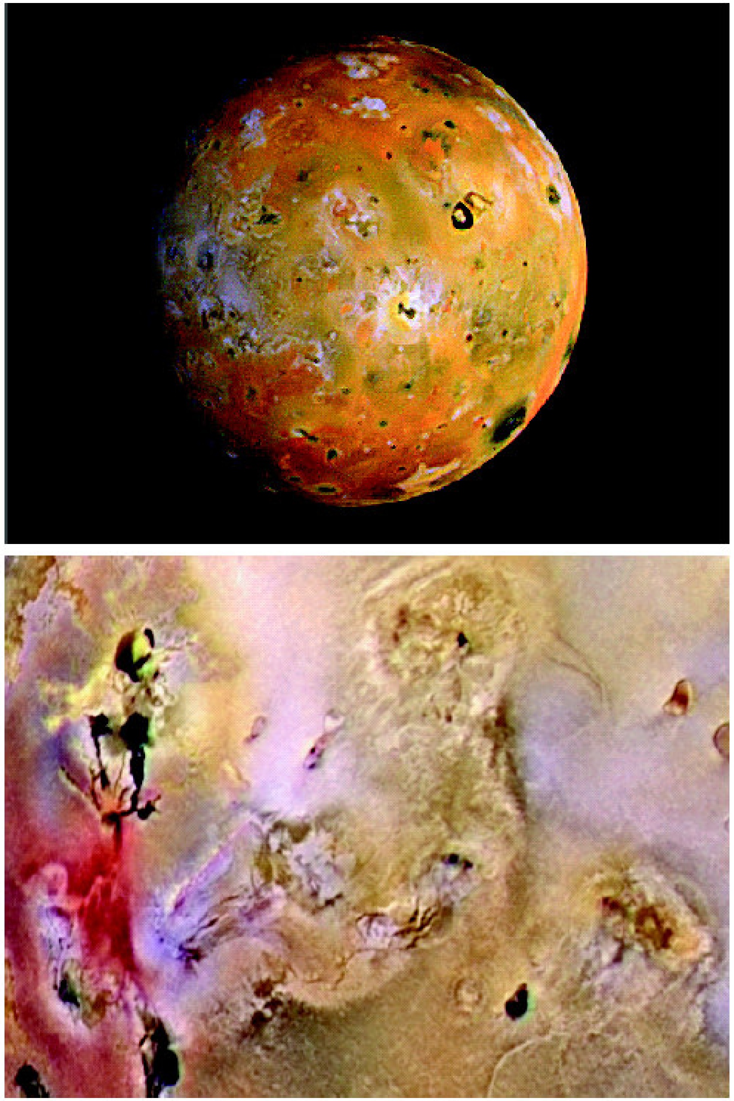
FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

more sophisticated color transformations can be used to combine grayscale images from different sensors as an example.



a b
c d
e f

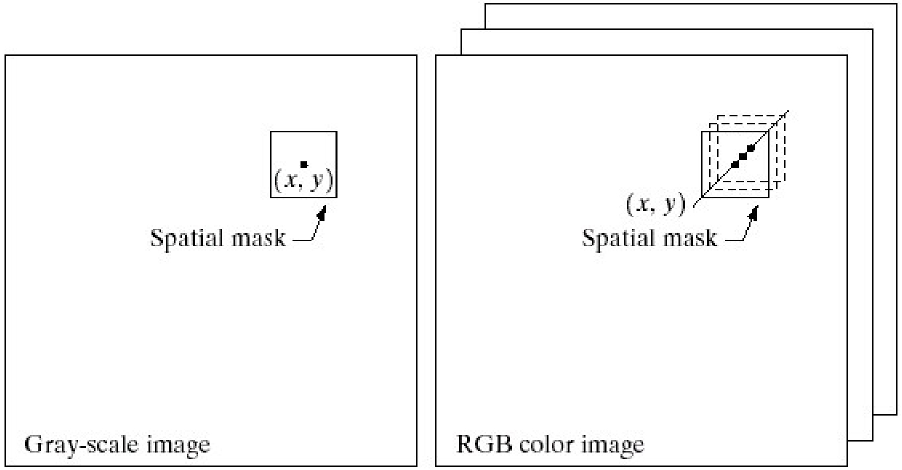
FIGURE 6.27 (a)–(d) Images in bands 1–4 in Fig. 1.10 (see Table 1.1). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (Original multispectral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



a
b

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

used a variety of different wavelengths
The newly ejected material is red (different material)
The older material is yellow. (sulfur)



a b

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

For neighborhood averaging operations are equivalent

Sum and divide all pixels in neighborhood

Sum and divide all the vectors in the neighborhood to get the same result as averaging each color component and averaging

$\boxed{\text{full-color image processing}}$

- process each component image separately and combine to form a composite image
- process color vectors (pixels) directly.

$$\underline{c}(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

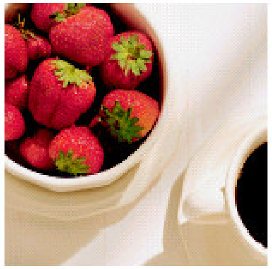

Full color

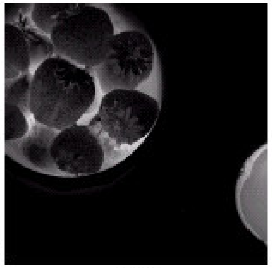
FIGURE 6.30 A full-color image and its various color-space components. (Original image courtesy of Med-Data Interactive.)

CMYK


Cyan


Magenta


Yellow


Black

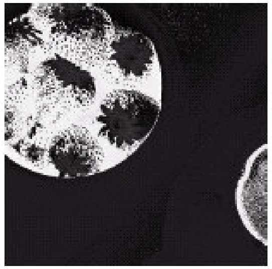
RGB


Red


Green


Blue

HSI


Hue


Saturation


Intensity

← just intensity

hue is difficult to interpret since its undefined for white, black and gray color transformations

$$\underbrace{s_i}_{\text{new color components}} = T_i(\underbrace{r_1, r_2, \ldots, r_n}_{\text{color components, i.e., R,G,B}})$$

$i = 1, \ldots, n = \#$ of color components

$T_i =$ set of color transformations

There are different costs associated with image processing in the different color spaces

to do intensity modification $g(x,y) = k f(x,y)$ $0 \le k \le 1$

HSI color space $s_3 = k r_3$

$s_1 = r_1$, $s_2 = r_2$

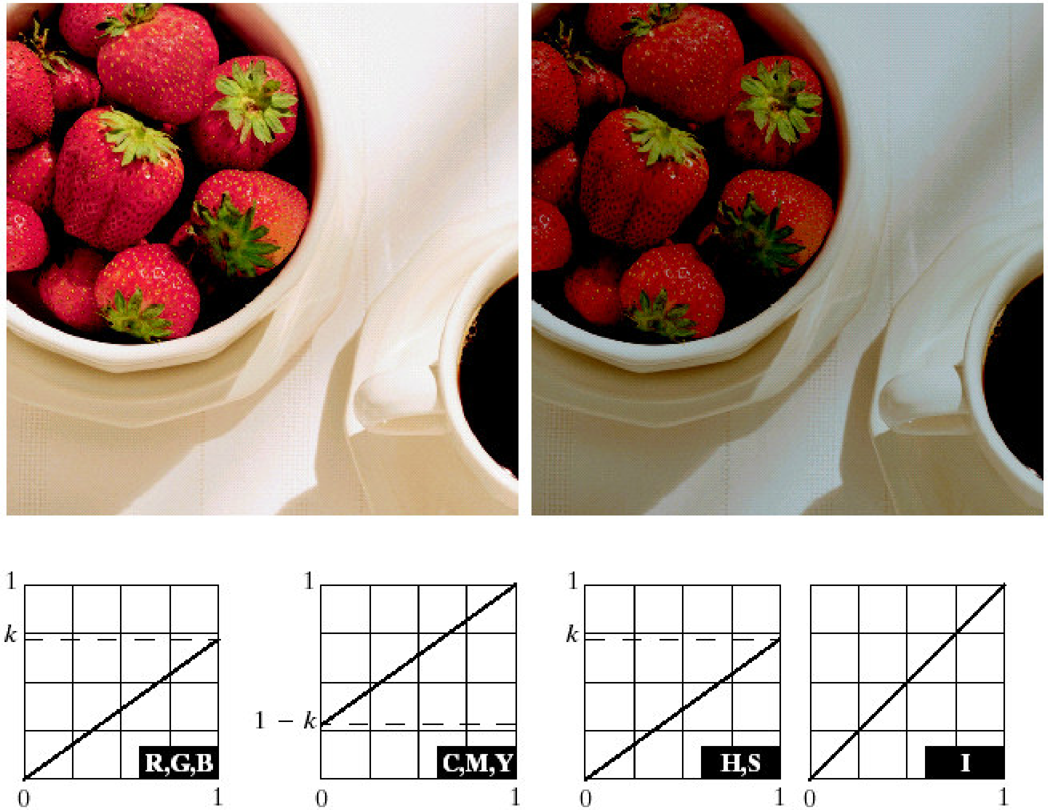
RGB color space $s_i = k r_i$, $i = 1, 2, 3$

CMY color space $s_i = k r_i + (1-k)$, $i = 1, 2, 3$

We didn't show it but $I = \frac{1}{3}\left[3 - (c+m+y)\right] = 1 - \frac{1}{3}(c+m+y)$
which is why this formula looks a little odd.

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.)



Scale each component in RGB

Since C=1−R, etc. This is simply a linear transformation.

these are reversed
I is decreased
H,S remain the same.

Color complements

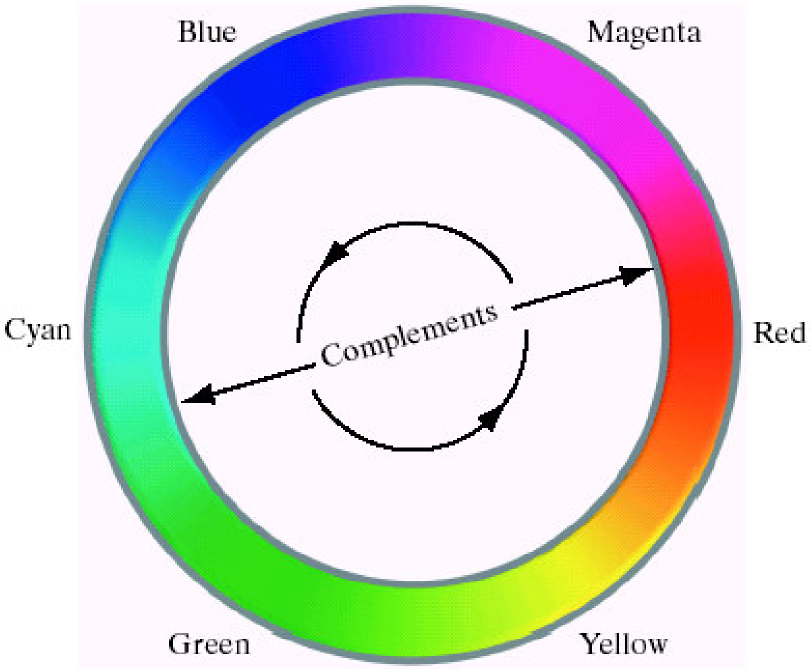
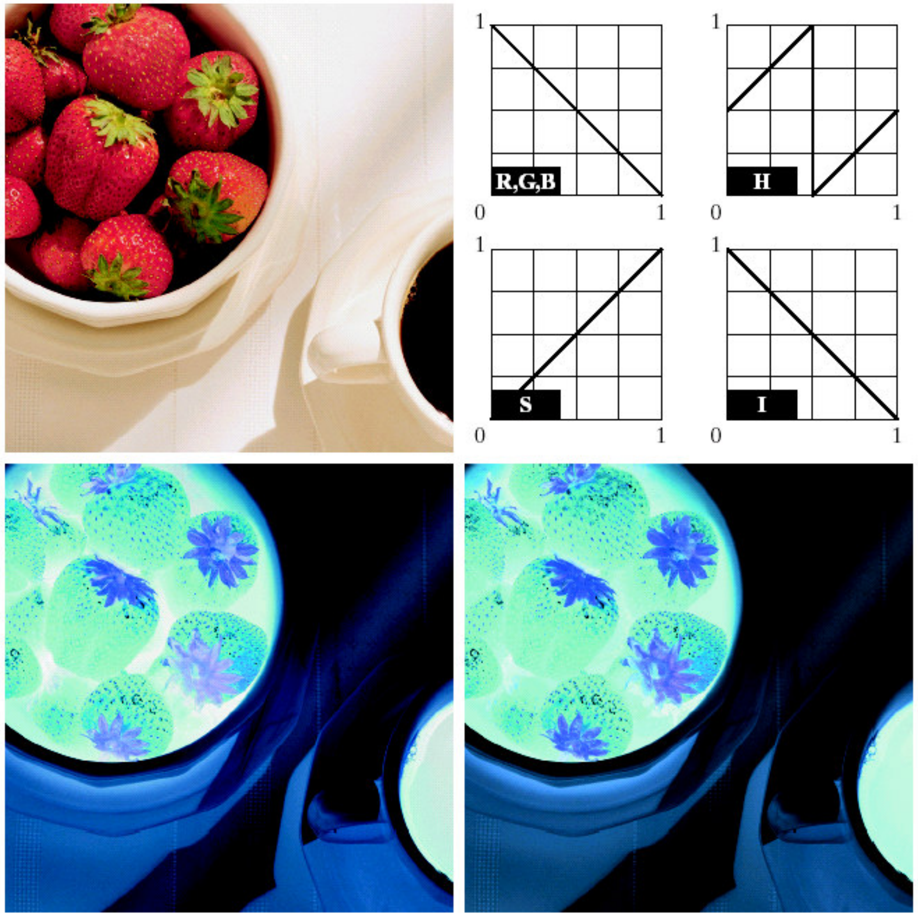


FIGURE 6.32 Complements on the color circle.

Newton's color circle summarizes the additive properties of colors



a b
c d

FIGURE 6.33 Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.

RGB complement

HSI complement.

As you decrease each color (R, G, B) its complement becomes evident.

R ⟷ cyan

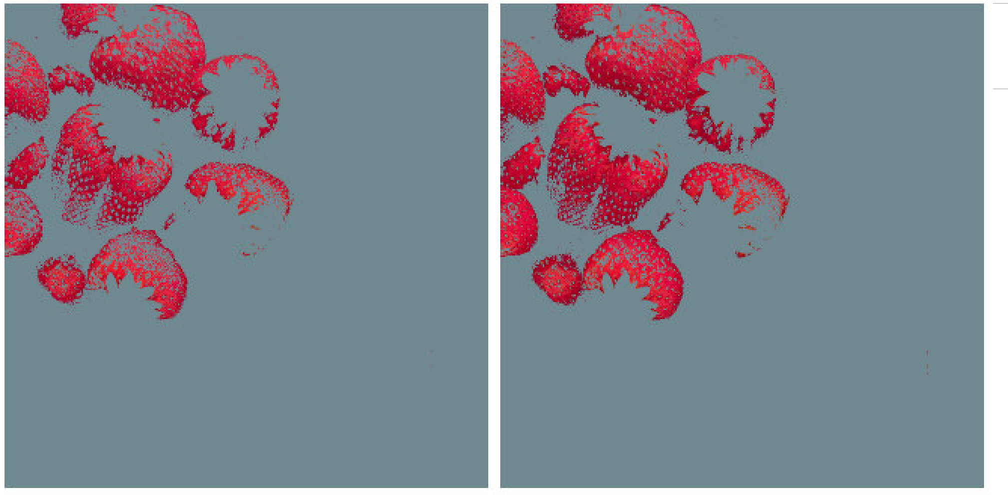
G ⟷ magenta

B ⟷ yellow

complement not straight forward
See problem 6.18

Color Slicing

RGB cube



RGB sphere

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 slicing - map colors outside a range of interest to a neutral color

neutral color

cube, hypercube

$$s_i = \begin{cases} 0.5 & \text{if } \left[|r_j - a_j| > \frac{W}{2}\right]_{1 \le j \le n} \\ r_i & \text{otherwise} \end{cases}$$

cube of width W centered at (a_1, a_2, a_3)

sphere

$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^{n} (r_j - a_j)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases}$$

$i = 1, 2, \ldots, n$

6.5.4. Tone/Color correction

need a device-independent color model to get color consistency between monitors & output devices

color management systems

Pantone (used by Adobe)

CIE L^*a^*b — similar to HSI by separating color from intensity

$$L^* = 116 \cdot h\left(\frac{Y}{Y_w}\right) - 16 \qquad \text{lightness}$$

$$a^* = 500\left[h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right)\right] \qquad \text{Red–Green}$$

$$b^* = 200\left[h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right)\right] \qquad \text{Green–Blue}$$

where
$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.787q + \frac{16}{116} & q \leq 0.008856 \end{cases}$$

X_w, Y_w, Z_w — reference white tristimulus.
· perfect diffused illuminated with CIE D65 light.
(this is defined to be daylight)

X, Y, Z — R, G, B tristimulus values

$L^*a^*b^*$ is

colorimetric — colors perceived as identical have identical values

perceptually uniform — color differences are perceived uniformly

device independent

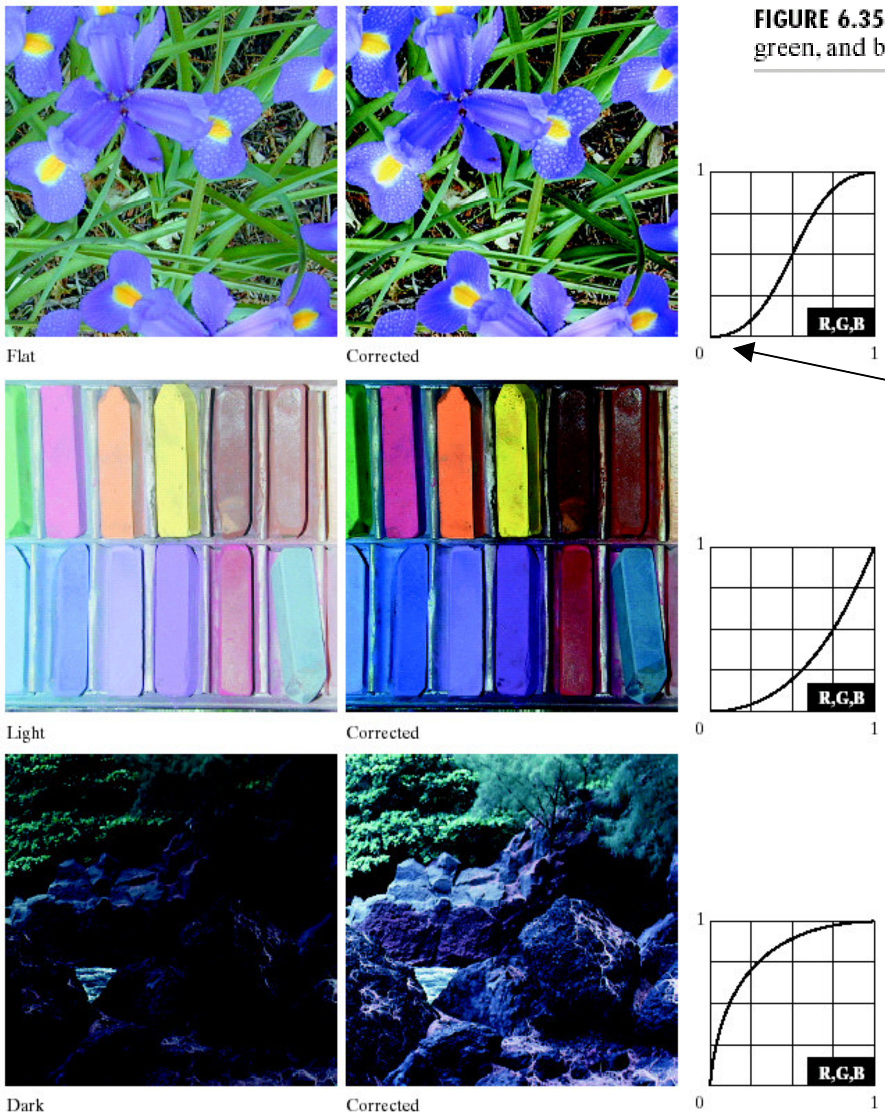


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

← lighten highlights

darken shadow areas

basically increases contrast

tonal range – key type

high-key – most information at high (bright) intensities

low-key – " " at low intensities

middle-key – " " at intermediate intensities

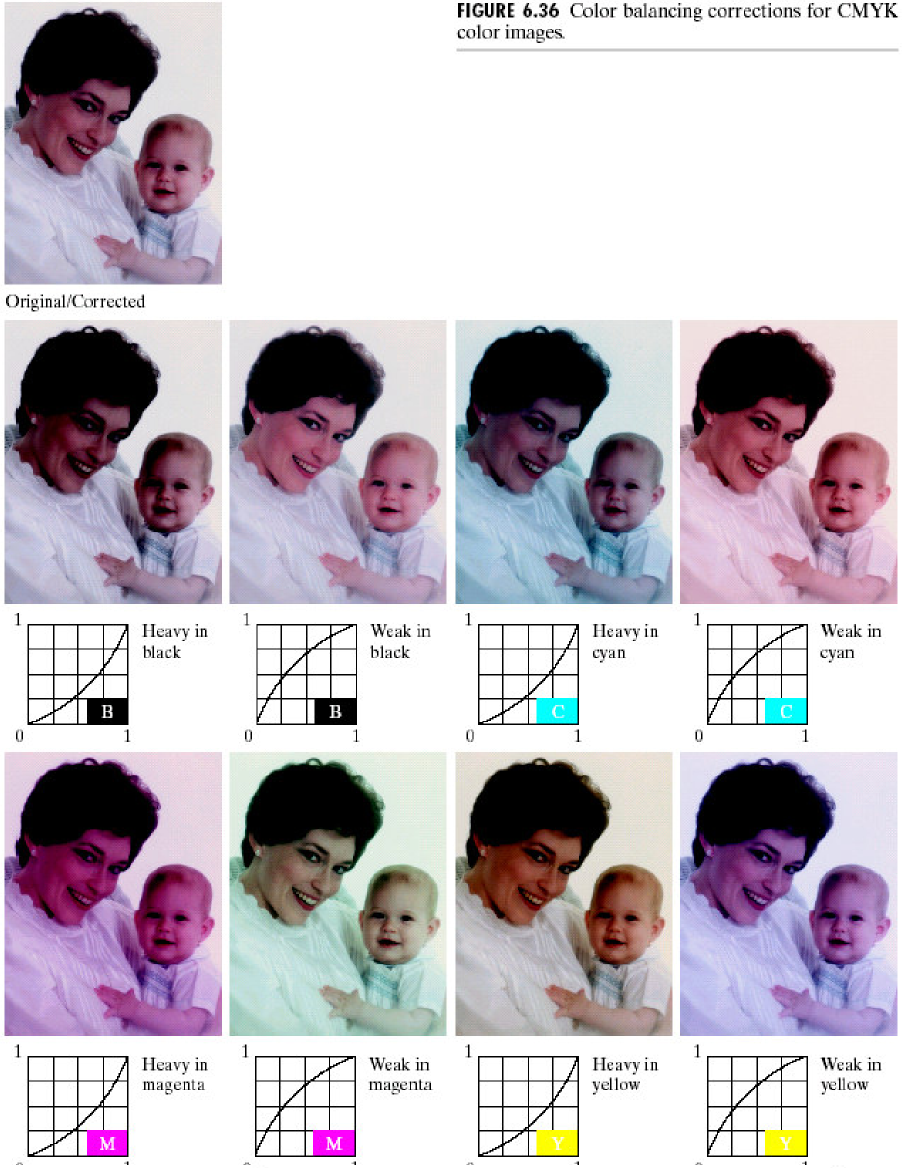
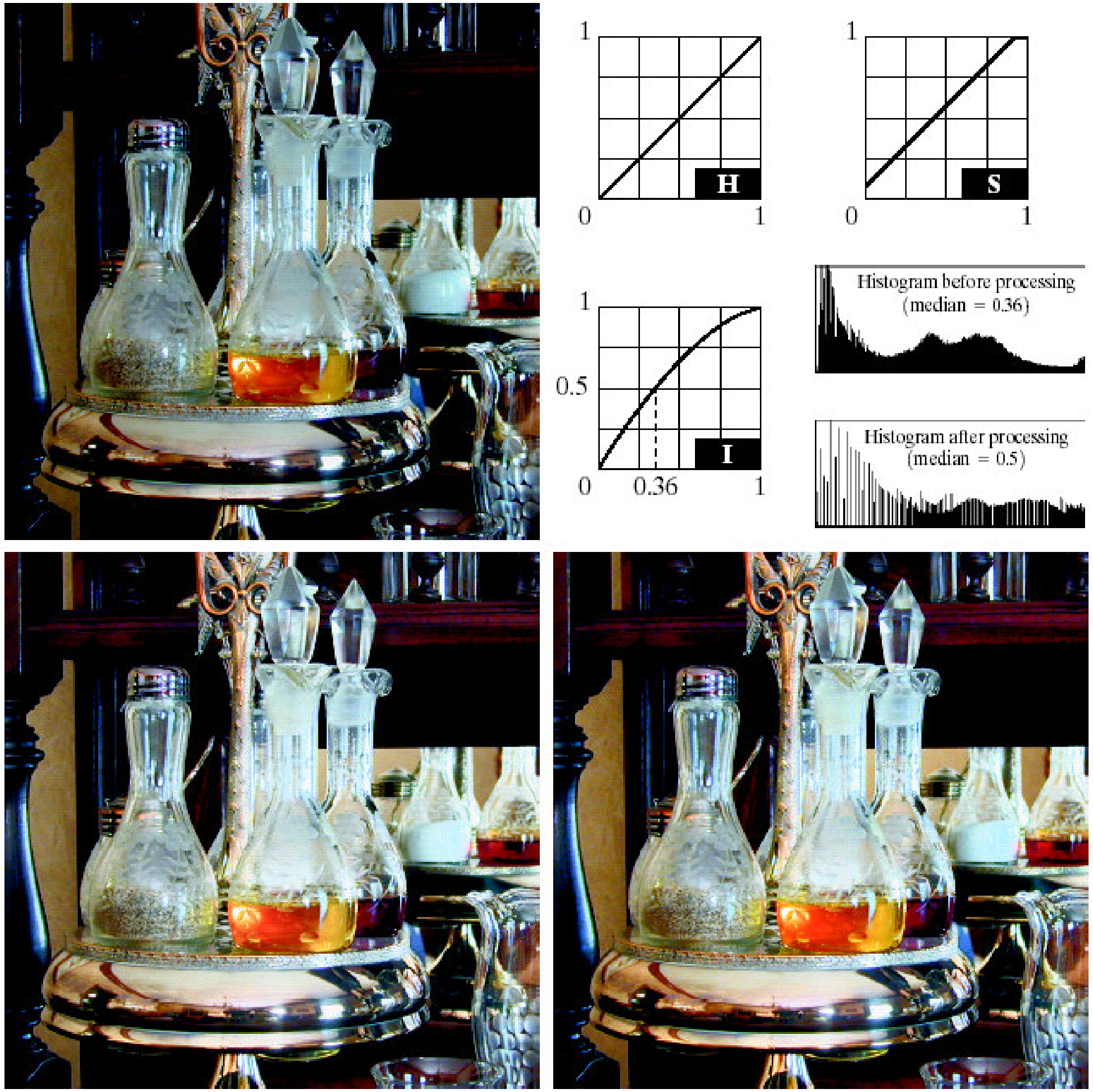


FIGURE 6.36 Color balancing corrections for CMYK color images.

Easiest way to evaluate color imbalance in an image is to analyze a known color. such as whites or skin.

Simple transformations to either boost or lighten a CMYK image.

Histogram Processing



a b
c d

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

Histogram equalization of intensity only

Increase image saturation slightly (after equalisation) to make colors look better

How can you apply histogram equalization to a color image?
Don't equalize colors independently,
spread color intensities such as in HSI space.

a b
c d

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

Color Image Smoothing

We just considered
pixel transforms

The next level of processing is
neighborhood processing
such as smoothing and sharpening

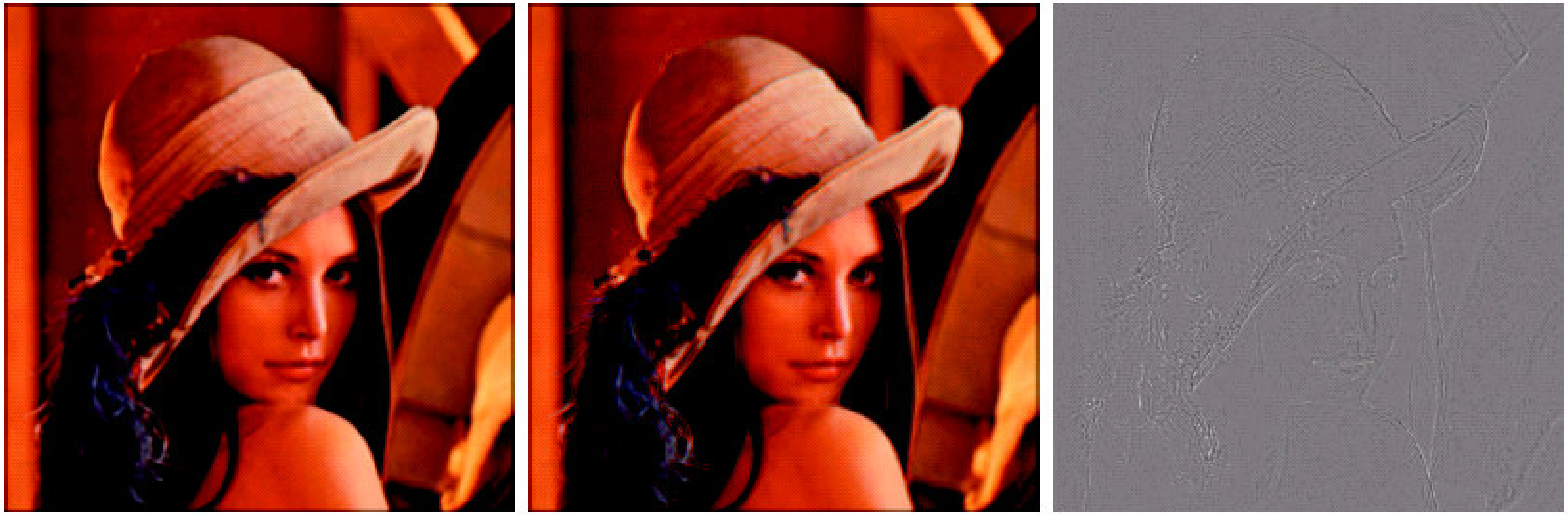
Consider averaging

$$\bar{c}(x,y) = \frac{1}{K} \sum_{(x,y) \in S_{xy}} \underline{c}(x,y) \qquad K = \# \text{ of pixels}$$

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

a b c

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



a b c

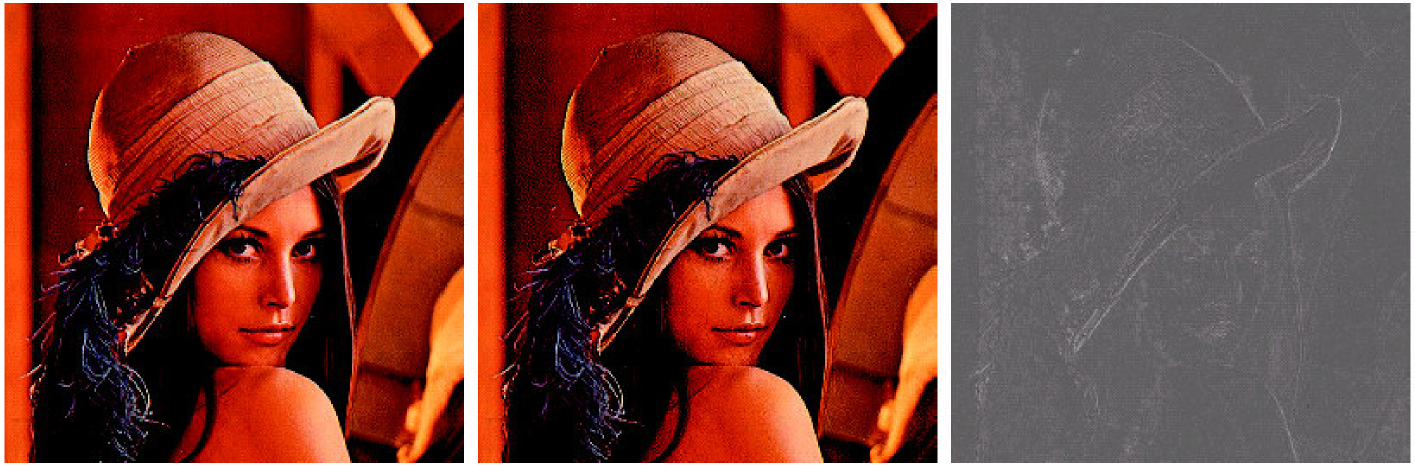
FIGURE 6.40 Image smoothing with a 5 × 5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

Image smoothing using a 5x5 mask.

(a) smoothing each color plane independently

(b) smoothing I (intensity) component of HSI image and conversion to RGB. This keeps color accurate.

(c) No data on how this difference image was computed. Several possibilities



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.

processing each RGB color plane

processing only HSI intensity plane

difference (?) image

$\boxed{\text{Color image sharpening using Laplacian}}$

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

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).]

Added Gaussian noise to the R, G, and B color planes.
RGB image with noise shown in (d).

a b c

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

significantly degraded due to non-linearity of cosine and min in transformations

intensity is average which tends to reduce noise.

HSI components of the noisy RGB image.

RGB image salt & pepper noise in Green channel

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.

noise spreads to all HSI component images

a b
c d

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

Compressed and decompressed using JPEG 2000

Slight blurring due to lossy technique