Lecture #13

- Point (pixel) transformations
  - Color modification
  - Color slicing
  - Device independent color
  - Color balancing
- Neighborhood processing
  - Smoothing
  - Sharpening
- Color segmentation

Color Transformations

Color transformations are similar to pixel or point transformations.

New color $s_i = T_i(r_1, r_2, \ldots, r_n)$ where $r_1, r_2, \ldots, r_n$ are the component colors and T_i is a <u>set</u> of transformations.

Color Transformations

In RGB color space:

$$I = \frac{1}{3}[R + G + B]$$

To modify intensity multiply each color component by k

$$I' = \frac{1}{3}[kR + kG + kB] = \frac{k}{3}[R + G + B] = kI$$

Color Transformations

In CMY color space intensity is:

$$I = \frac{1}{3}\left[(1-C)+(1-M)+(1-Y)\right] = \frac{1}{3}\left[3-C-M-Y\right] = 1-\frac{1}{3}\left[C+M+Y\right]$$

To modify intensity modify each component as: $C' = kC + (1-k)$

$$I' = 1-\frac{1}{3}\left[kC+(1-k)+kM+(1-k)+kY+(1-k)\right]$$

$$= 1-\frac{1}{3}\left[k(C+M+Y)+3(1-k)\right] = 1-\frac{k}{3}\left[C+M+Y\right]-\frac{3}{3}\left[1-k\right]$$

$$= 1-\frac{k}{3}\left[C+M+Y\right]-1+k = k\left[1-\frac{1}{3}(C+M+Y)\right] = kI$$

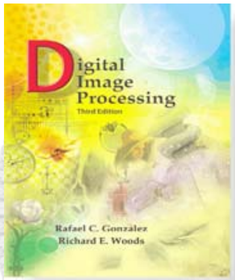
Color Transformations

The cost of performing transformations will vary depending upon the color space you are using.

Consider an intensity transformation g(x,y)=kf(x,y) for 0≤k≤1

- In HSI color space: $s_1=r_1$, $s_2=r_2$, $s_3=kr_3$
- In RGB color space: $s_i=kr_i$, k=1,2,3
- In CMY color space: $s_i=kr_i+(1-k)$, k=1,2,3

$$I=\frac{1}{3}[R+G+B]=\frac{1}{3}\big[(1-C)+(1-M)+(1-Y)\big]=1-\frac{1}{3}[C+M+Y]$$



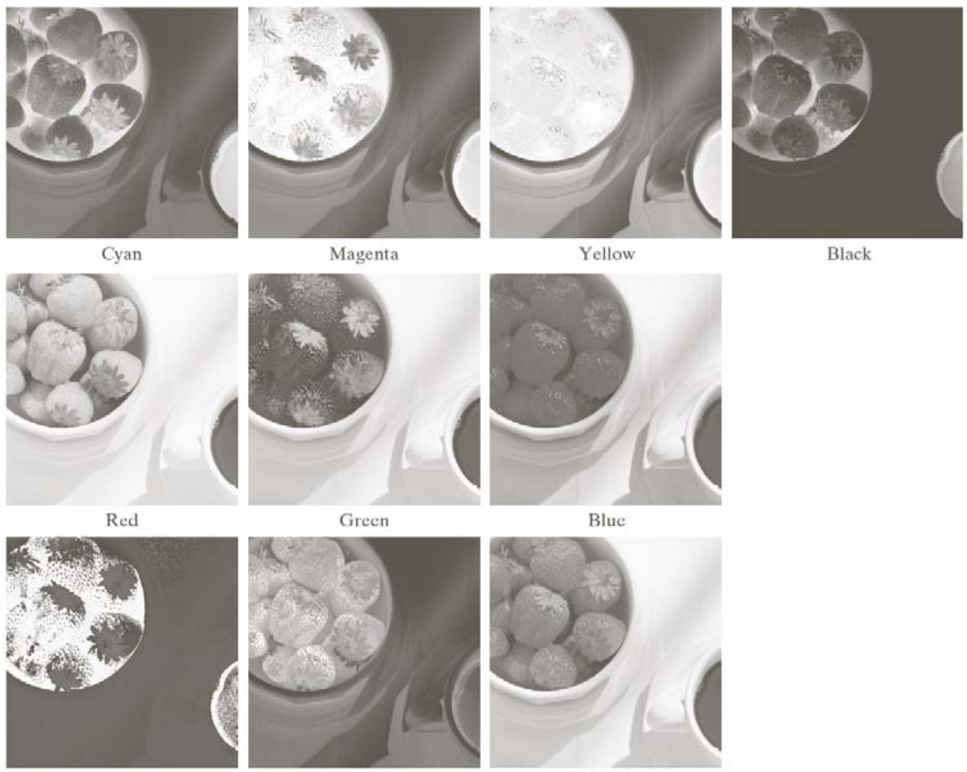
Color Transformations

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

CMYK

RGB

HSI



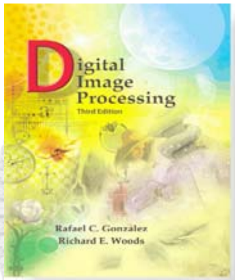
Remember Y absorbs B and transmits R & G

Hue is difficult to interpret for black, white and any gray since it is undefined.

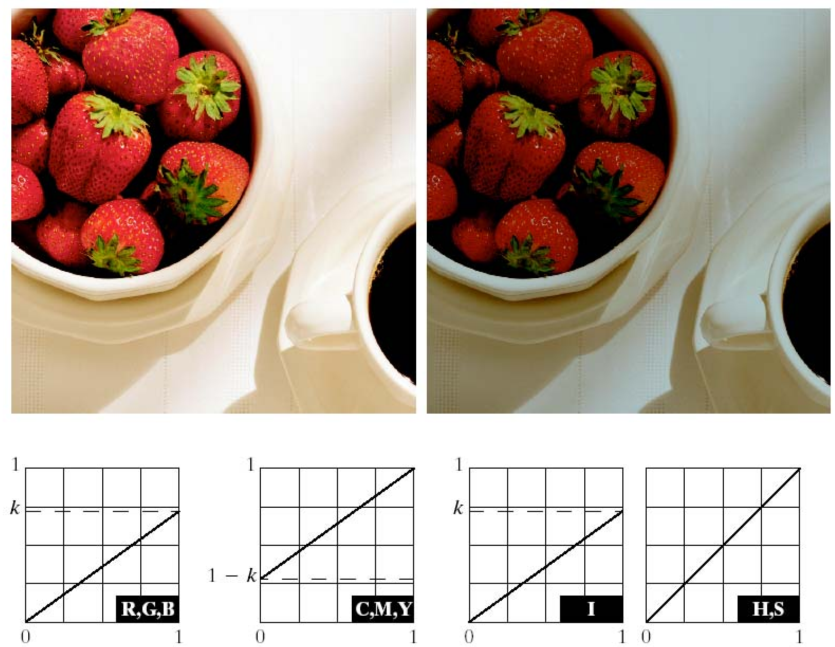
Strawberries are highly saturated so they are white

Just intensity

$$I=\frac{1}{3}[R+G+B]=\frac{1}{3}[(1-C)+(1-M)+(1-Y)]=1-\frac{1}{3}[C+M+Y]$$



Intensity Transformation



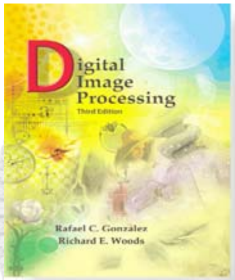
a b
c d e

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 RGB component

C=1-R;
M=1-G;
Y=1-B

I decreases since R,G,B decrease.
H and S do not change.
(This slide corrected since last edition)



Color complements

cool colors

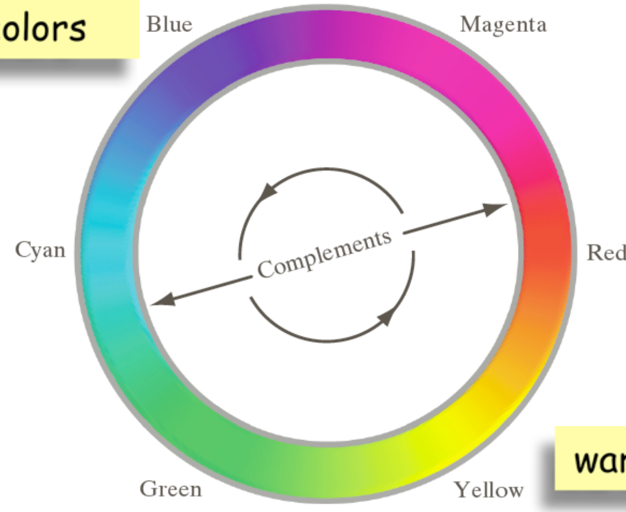
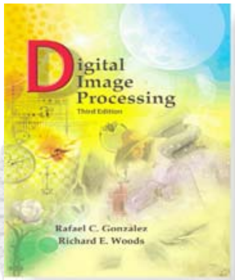


FIGURE 6.32 Complements on the color circle.

warm colors

Newton's color circle summarizes complementary colors.

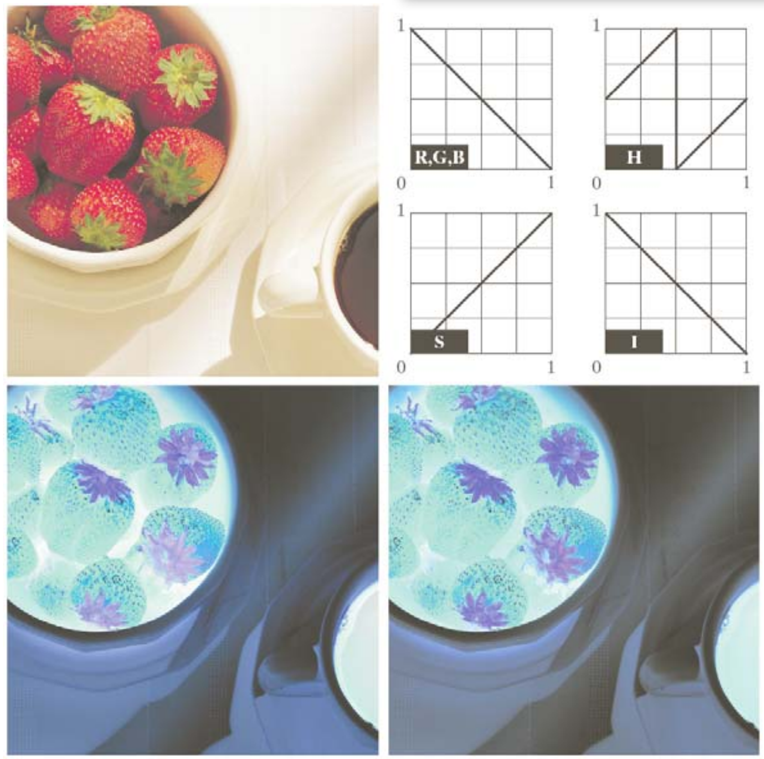
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



Complement Transforms

RGB complement simply 1-R, 1-G, 1-B

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



a b
c d

FIGURE 6.33 Color complement transformations. (a) Original image. (b) Complement transformatic functions. (c) Complem of (a) based c the RGB map functions. (d) approximatio of the RGB complement using HSI transformatic

RGB complement

HSI "approximate" complement

I goes down because R,G,B are going down.

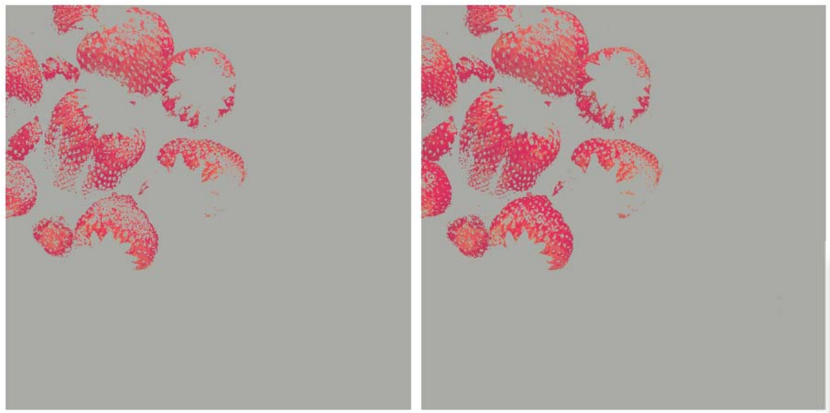
S does not change.

H complement transform not obvious. (see problem 6.18) Basically colors undergo a 180° angle shift

Color Slicing Transforms

$$s_i = \begin{cases} 0.5 & \text{if } \left| r_j - a_j \right| > \dfrac{W}{2} \\ r_i & \text{otherwise} \end{cases}$$

(0.5,0.5,0.5) defines a neutral gray outside a RGB cube centered at (0.6863,0.1608,0.1922)

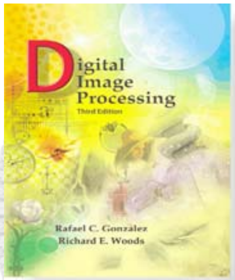


$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^{n} \left(r_j - a_j \right)^2 > R_o^2 \\ r_i & \text{otherwise} \end{cases}$$

(0.5,0.5,0.5) defines a neutral gray outside a RGB sphere centered at (0.6863,0.1608,0.1922)

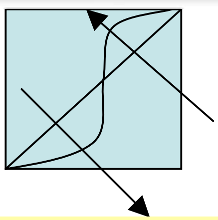
FIGURE 6.34 Color-slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at (0.6863, 0.1608, 0.1922), and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5).

Sphere might be slightly better at fully capturing area of strawberries.



Color Intensity Transforms

Make brights brighter

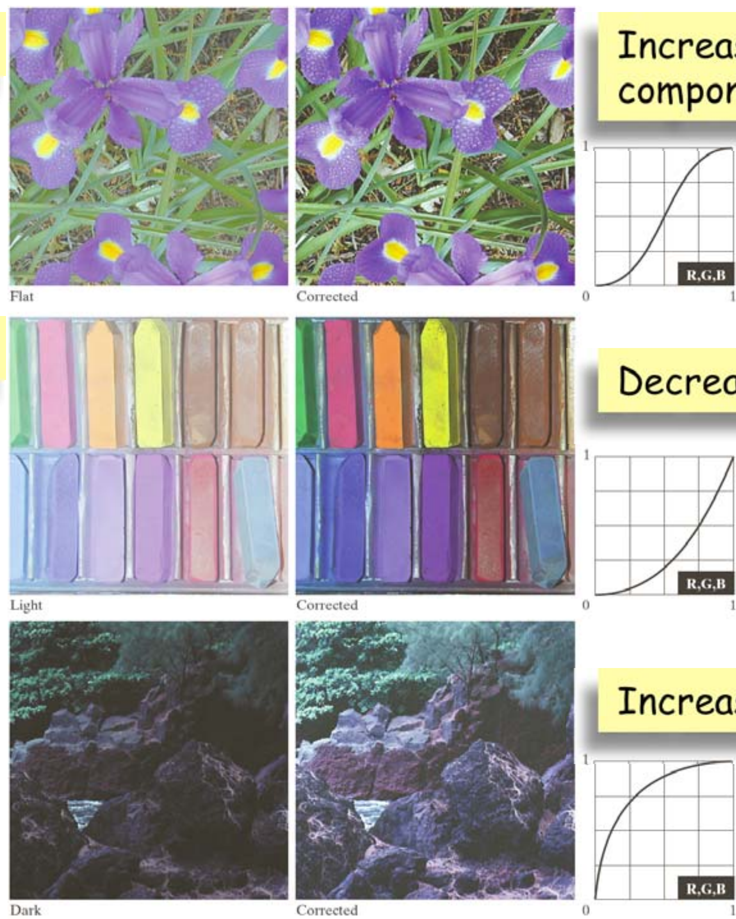


Make darks darker when applied uniformly to R,G, and B

flat

light

dark



Increase contrast of each component at high and low ends

Decreases brightness overall

Increases brightness overall

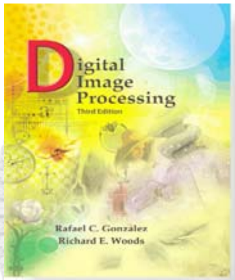
FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

Color Transforms

We need a device independent color model to get consistency between monitors and printers

L*a*b* is

- Colorimetric - colors perceived as identical have identical values
- Perceptually uniform- color differences are perceived uniformly
- Device independent



Color Transforms

L*a*b* is similar to HSI in that it separates color from intensity: lightness, R-G, G-B

X,Y,Z are the RGB tristimulus values of a color

X_w,Y_w,Z_w are the RGB tristimulus values of a perfect white diffuser illuminated with a CIE D65 (daylight) lamp

Color Transforms

Lightness

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

Red-Green

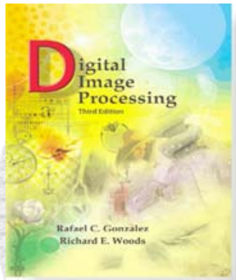
$$a^* = 500\left[h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right)\right]$$

Green-Blue

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

where

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



Color Balancing

The easiest way to detect a color imbalance in an image is to analyze a known color such as whites or skintone.

Correct using simple transformation to boost or lighten a particular color.

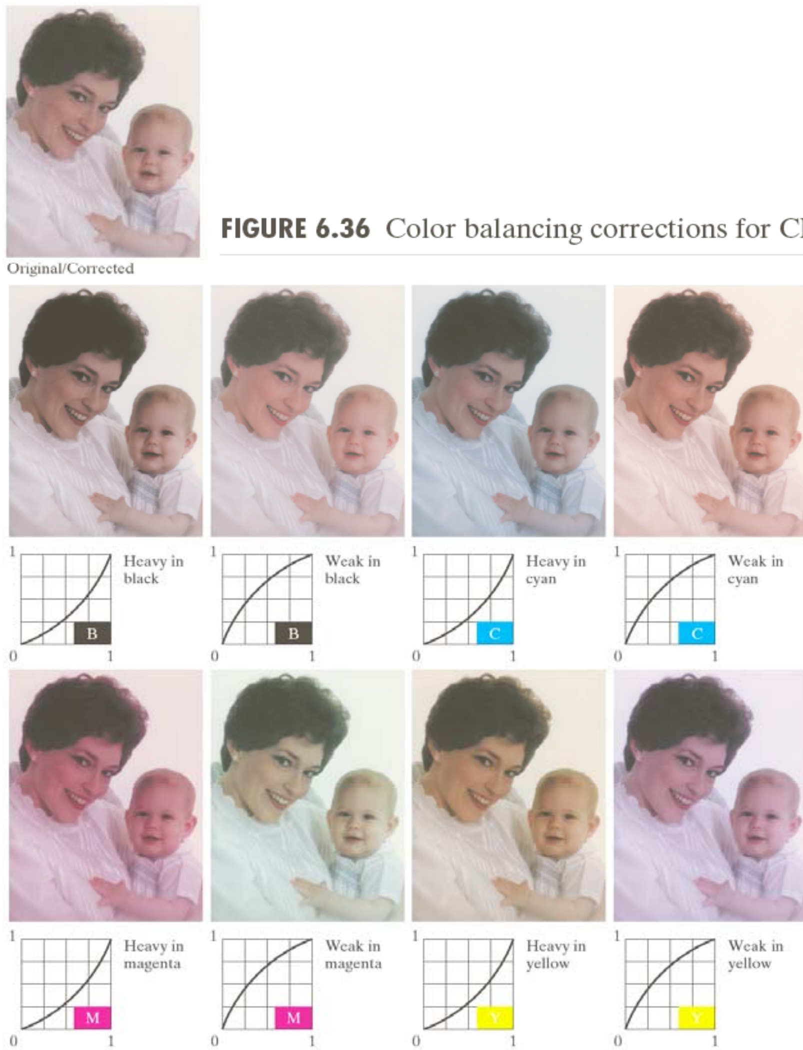
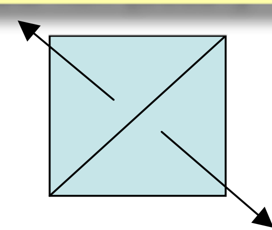
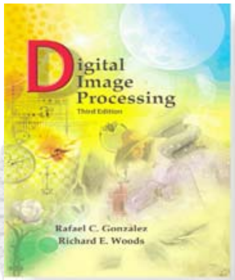


FIGURE 6.36 Color balancing corrections for CMYK color images.

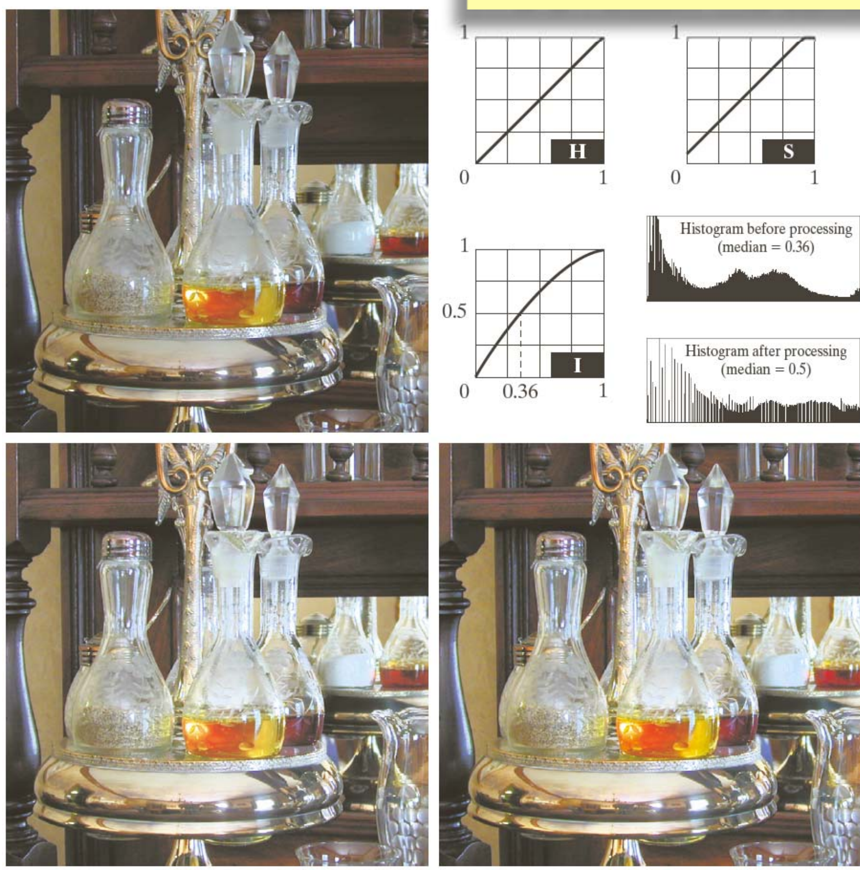




Color Histogram Equalization

H & S remain the same.

Original image.



a b
c d

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

Equalize the intensity.

Image after equalization of intensity.

Increase saturation slightly (after equalization) to make colors look better.

How can you equalize a color image? Use a color space such as HSI which separates intensity from color so you can equalize intensity.

Lena: RGB space

a b
c d

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

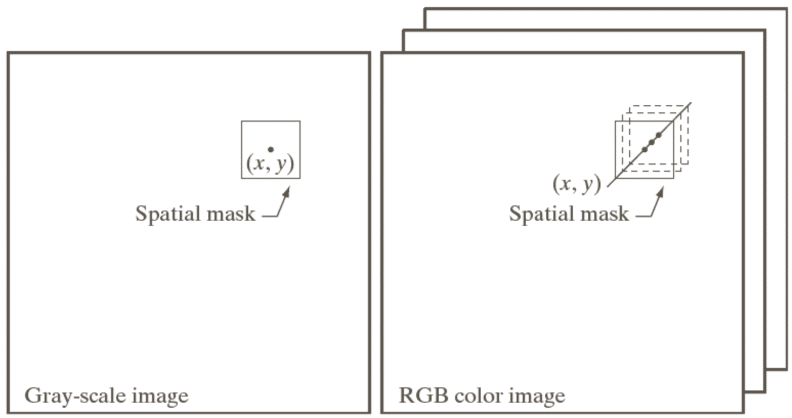
Lena: HSI space

a b c

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

Color Neighborhood Processing

1. Combine each component image separately and combine into one composite image
2. Process color vectors directly

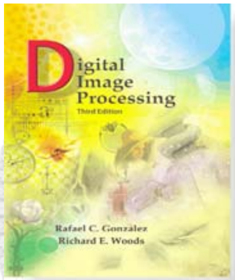


a b

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

Averaging will give the same result using either method.

1. Sum all pixels and average in each color plane. Then combine via averaging.
2. Compute average (mean) of all vectors in the neighborhood.



Color Neighborhood Transforms

The color transforms considered to know were point (or pixel) transforms.

The next level of processing is neighborhood processing such as smoothing or sharpening.

Color averaging:

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

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

Color Smoothing

Using a 5x5 uniform mask

Smoothing RGB color planes independently.

Smooth only I in HSI and convert back to RGB.

Subtract images*

Average of two pixels of different colors is yet another different color. Averaging in I does not change color.

Minimizes color shifts

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.

*no details of how computation was performed given

Color Sharpening

Sharpening $\quad g(x,y) = f(x,y) \pm \nabla^2 f(x,y)$

Use a vector approach with Laplacian:

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

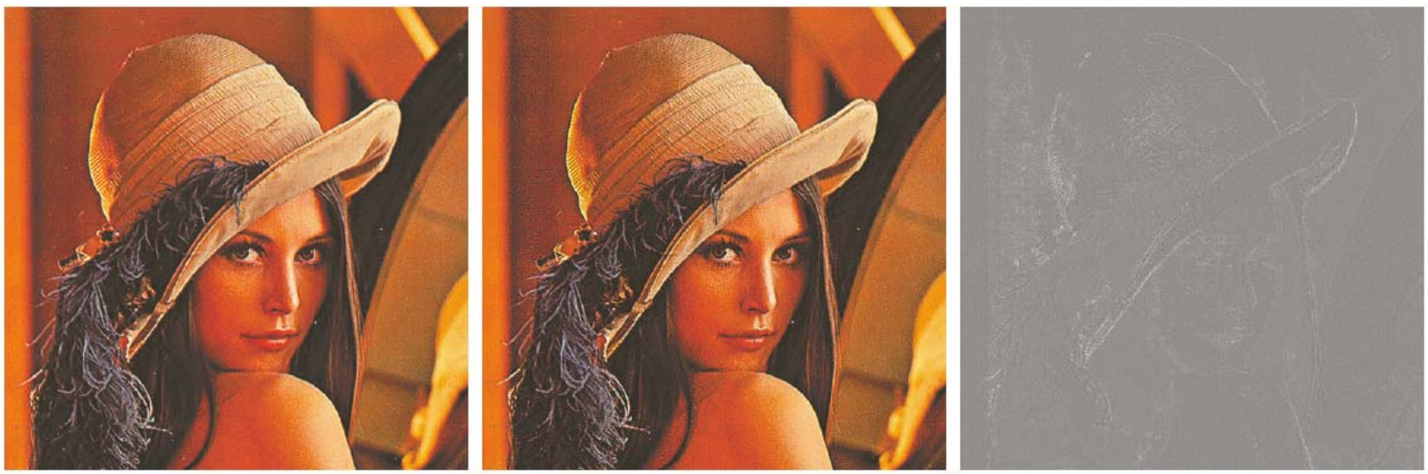
Color Sharpening

Using a Laplacian

Sharpen RGB color planes independently.

Sharpen only I in HSI and convert back to RGB.

Subtract images*

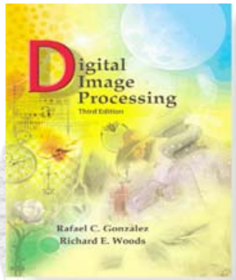


Minimizes color shifts

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.

*no details of how computation was performed given



Color Image Segmentation

Divides images into constituent regions or objects

Usually works better in RGB space than HSI space

Objective is to classify each color pixel as having a color within the specified range or not

Use a distance classifier

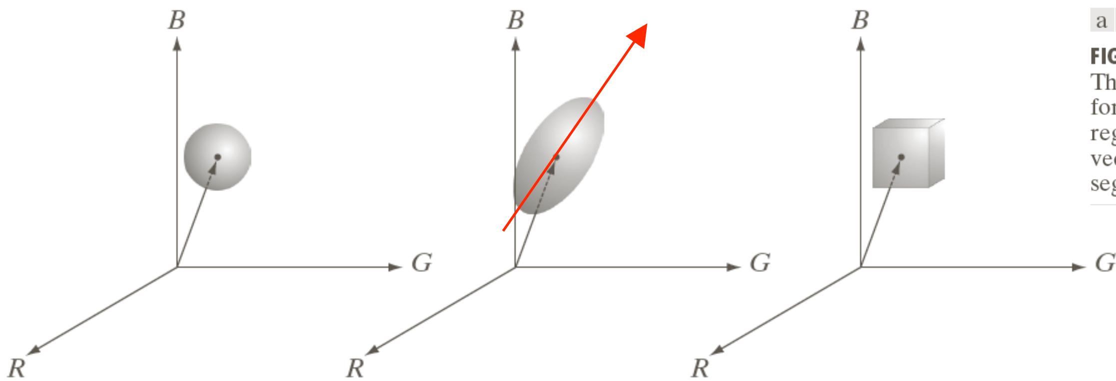
$$D(\vec{z},\vec{a}) = \|\vec{z}-\vec{a}\| = \sqrt{(\vec{z}-\vec{a})^T(\vec{z}-\vec{a})} = \sqrt{(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2}$$

More generally

$$D(\vec{z},\vec{a}) == \sqrt{(\vec{z}-\vec{a})^T C^{-1}(\vec{z}-\vec{a})}$$

(This is really color slicing since no connectivity)

Color Image Segmentation



a b c

FIGURE 6.43 Three approaches for enclosing data regions for RGB vector segmentation.

Spherical using Euclidian distance

Ellipsoidal happens when you use the covariance matrix C

Box is computationally very effective since no squares or square roots. Choose sides proportional to standard deviations of each color.

HSI Color Image Segmentation

HSI is the most natural color space in which to do color segmentation (a Ch.10 topic)

We want to segment the pixels in this red area.

H image

S image

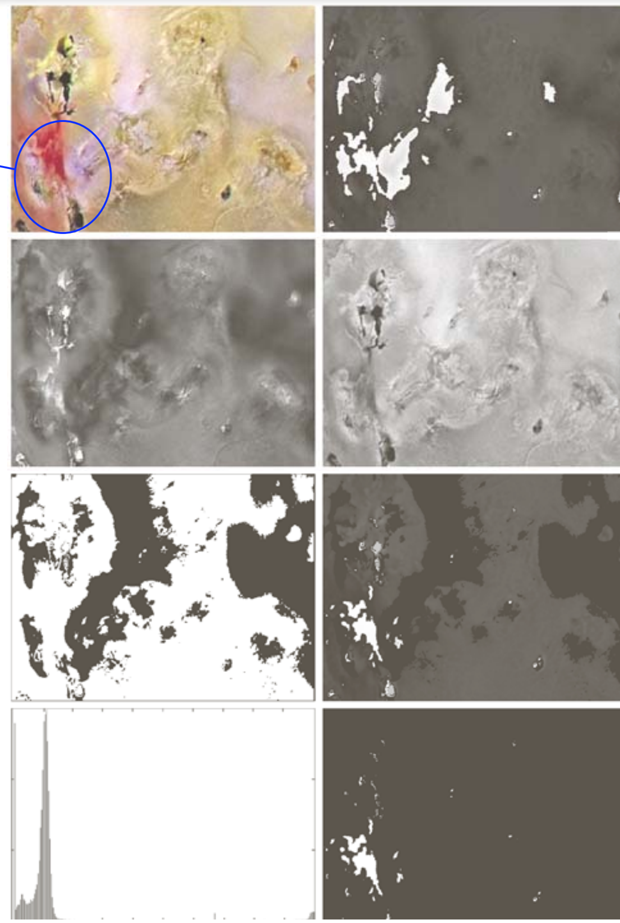
I image

1. Threshold S image at 0.1 of max_sat_value to get saturated color mask.

2. Multiply thresholded S image by H image to identify regions of saturated color

3. Histogram of masked H image.

4. Threshold masked image at 0.9 to get only pixels near red color



a b
c d
e f
g h

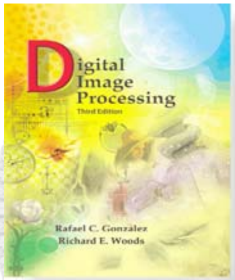
FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).

RGB Color Image Segmentation

1. Select a sample of region to be segmented
2. Compute mean and standard deviations of the color contained in the sample rectangle
3. Construct box in color space corresponding to standard deviations of sample

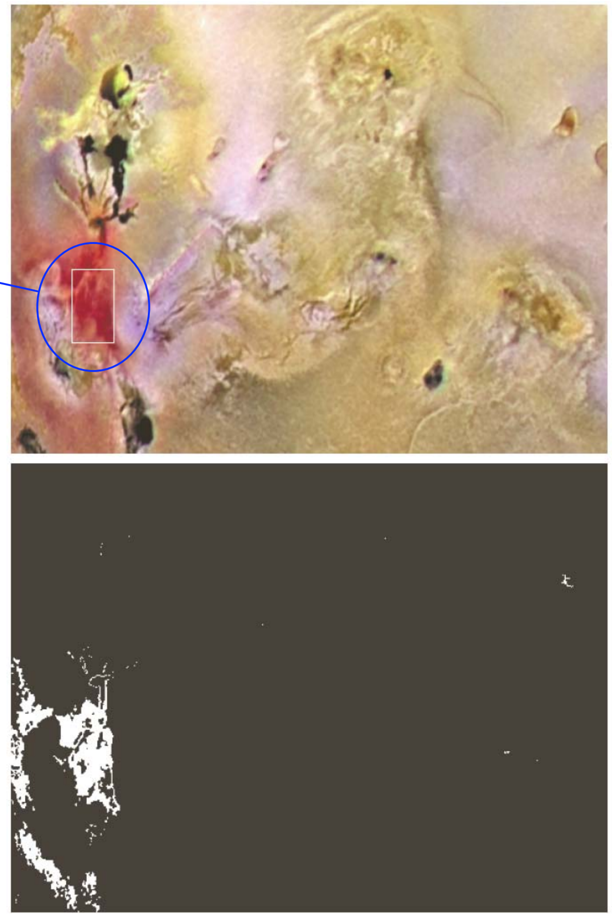
$$a_R \pm 1.25\sigma_R$$
$$a_G \pm 1.25\sigma_G$$
$$a_B \pm 1.25\sigma_B$$

4. Classify pixel as white if inside this box else black



RGB Color Image Segmentation

Rectangle shows color we want to segment



a
b

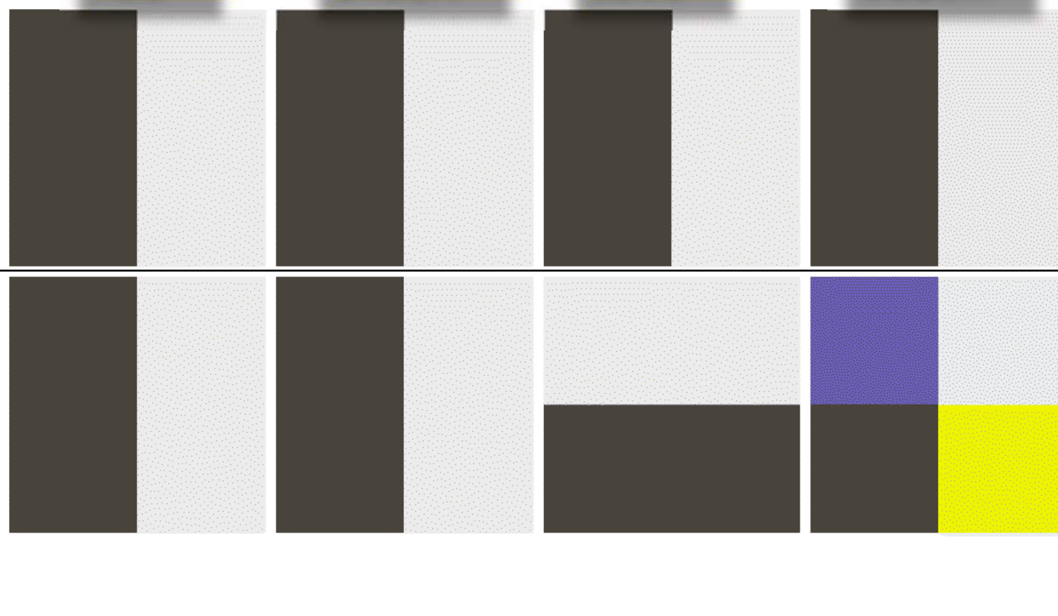
FIGURE 6.44 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 6.42(h).

Color Edges

RED GREEN BLUE COLOR

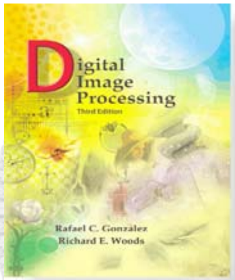
Color edges aligned

Color edges NOT aligned



If we simply added the magnitudes of the gradients they would be the same at the center. Intuitively they cannot be the same.

FIGURE 6.45 (a)–(c) *R*, *G*, and *B* component images and (d) resulting RGB color image. (e)–(g) *R*, *G*, and *B* component images and (h) resulting RGB color image.



(Vector) Color Gradient

See Section 6.6 and p. 563-564 of GWE. Implemented as colorgrad.

The maximum rate of change of a color vector c(x,y) at (x,y) is given by

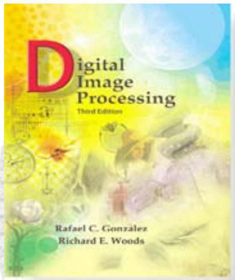
$$F(\theta)=\sqrt{\frac{1}{2}\left(g_{xx}+g_{yy}\right)+\left(g_{xx}-g_{yy}\right)\cos 2\theta+2g_{xy}\sin 2\theta}$$

in the direction

$$\theta(x,y)=\frac{1}{2}Tan^{-1}\left[\frac{2g_{xy}}{g_{xx}-g_{yy}}\right]$$

NOTE: This expression gives two directions. One is the direction of the maximum of F; the other is the direction of the minimum.

S.D.Zenzo, "A Note on the Gradient of a Multi-Image," Computer Vision, Graphics and Image Processing, Vol. 33, pp.116-125, 1986.



(Vector) Color Gradient

Let $\hat{r}, \hat{g}, \hat{b}$ be the unit vectors along the RGB axes of a RGB color space. Define

$$\vec{u} = \frac{\partial R}{\partial x}\hat{r} + \frac{\partial G}{\partial x}\hat{g} + \frac{\partial B}{\partial x}\hat{b} \qquad \vec{v} = \frac{\partial R}{\partial y}\hat{r} + \frac{\partial G}{\partial y}\hat{g} + \frac{\partial B}{\partial y}\hat{b}$$

Further define

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

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

$$g_{xy} = \vec{u} \bullet \vec{v} = \vec{u}^T \vec{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}$$