Outline

- Pseudo-coloring via
  - intensity slicing
  - color transformations, and
  - Multispectral imaging
- Full-color image processing
  - Intensity and color component manipulation
  - Contract enhancement and tonal correction via histogram equalization
  - Color image smoothing and sharpening
  - Color segmentation
  - Computing gradients
  - Noise filtering

A quick review of color spaces

- Humans can distinguish about 30-50 different shades of grey, but thousands of different color shades
- Also for automatic methods, color is important (several different colors can have the same intensity)
- So why did we spend most of the course on grey-level images?
    - The same methods can be applied
    - The key is to understand the color models!

To which color space (RGB or HSI) would you connect the following claim?

- Repsesents colors with red, green, and blue color channels?

  RGB

- Separates colors into hue, saturation, and intensity?

  HSI

- Practical for human description of colors?

  HSI

- Easy and straightforward?

  RGB

- Not device independent?

  HSI and RGB

To which color space (RGB or HSI) would you connect the following claim?

- Difficult transformations (singularities)?

  HSI

- Intensity decoupled?

  HSI

- Suits hardware implementation?

  RGB

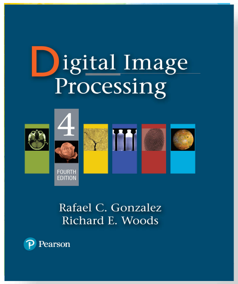
- Mimics the human color perception?

  RGB (from the point of view of the three types of cones we have in the retina whose responses peak at Red, Green and Blue wavelengths), but

  HSI is the one that really mimics how humans perceive and identify colors.

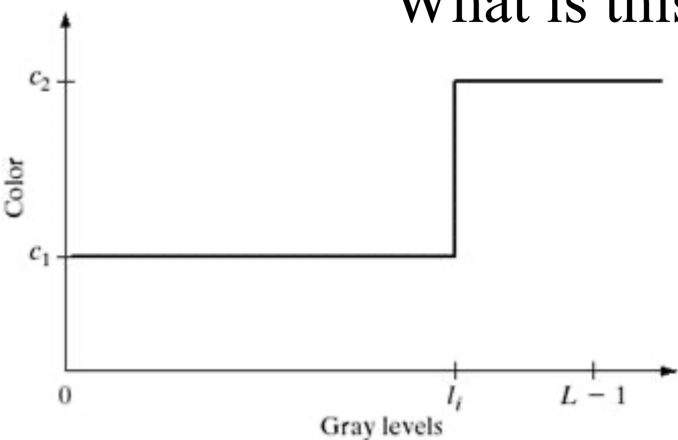
- Difficult to display?

  HSI

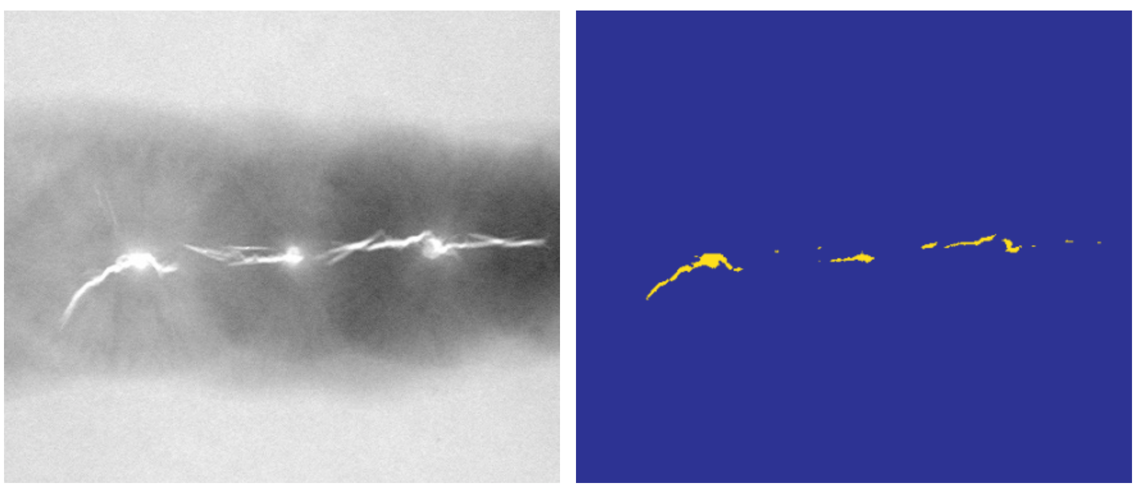


Pseudocoloring

What is this transformation doing to the image?



- Intensity slicing
- An example of pseudocoloring

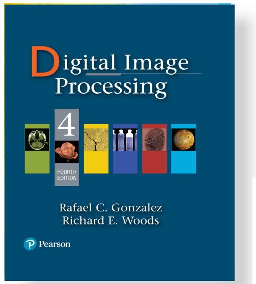


a b

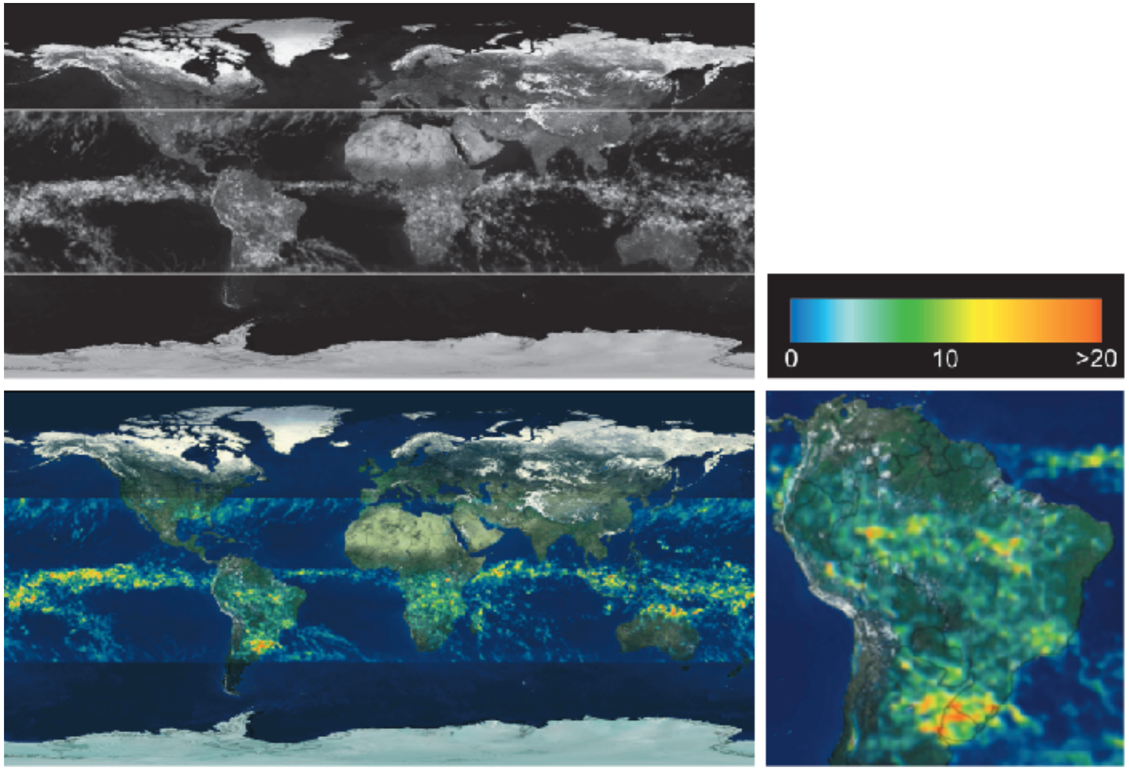
FIGURE 7.19 (a) X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)

Pseudocoloring

- Humans can distinguish about 30-50 different shades of grey, but 100,000 – 10,000,000 colors
- It is useful to display gray scale images using color to visualize the information better
- Color can also be used to represent some physical properties
    - Important to include color info to understand what they represent



Another pseudocoloring example: monthly rainfall

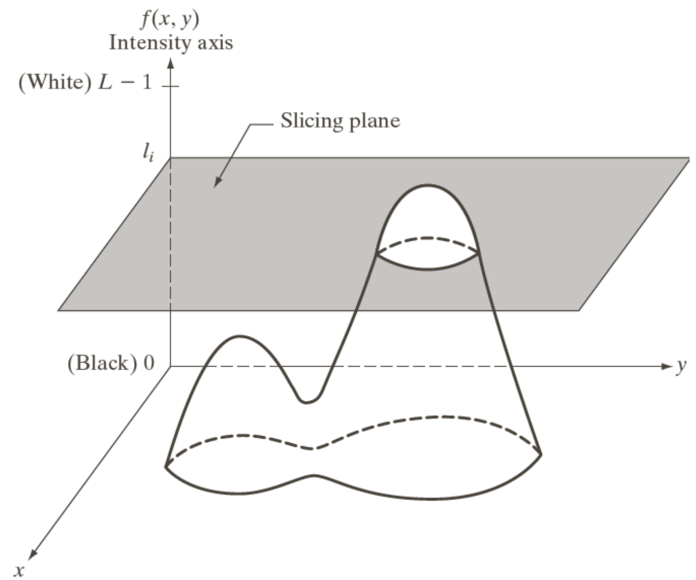


a b
c d

FIGURE 7.20 (a) Grayscale image in which intensity (in the 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.)

Intensity Slicing

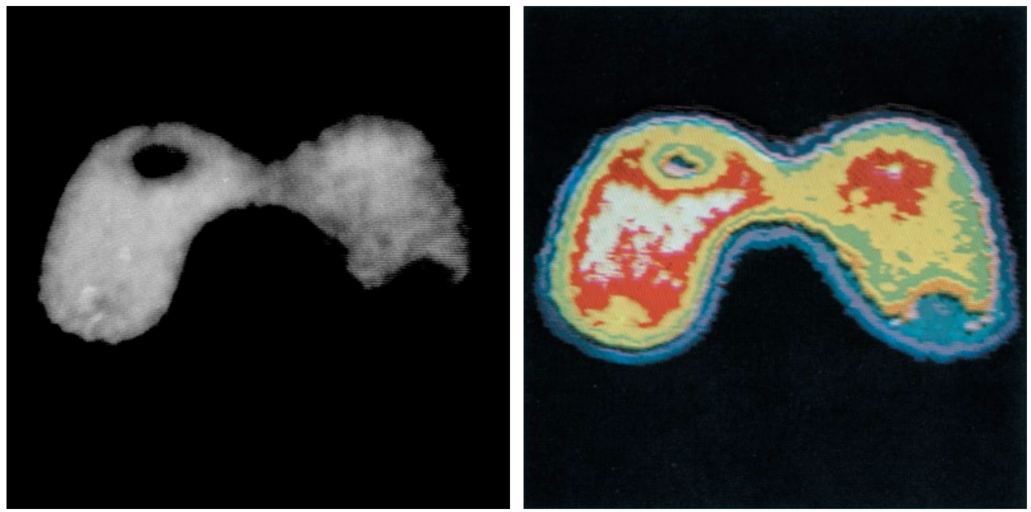
- Assign different colors to levels above and below the slicing plane
- Usually, several levels are used
- Intensity slicing can be thought of as another form of quantization in which the quantization levels get different colors



Intensity Slicing for Pseudocoloring

a b

FIGURE 7.18 (a) Grayscale image of the Picker Thyroid Phantom. (b) Result of intensity slicing using eight colors. (Courtesy of Dr. J. L. Blankenship, Oak Ridge National Laboratory.)



A simple example is shown below, a grey-scale image of the Picker Thyroid Phantom (a radiation test pattern). It is difficult to see the details of gray level variations in some areas.

Pseudocoloring with color transformations

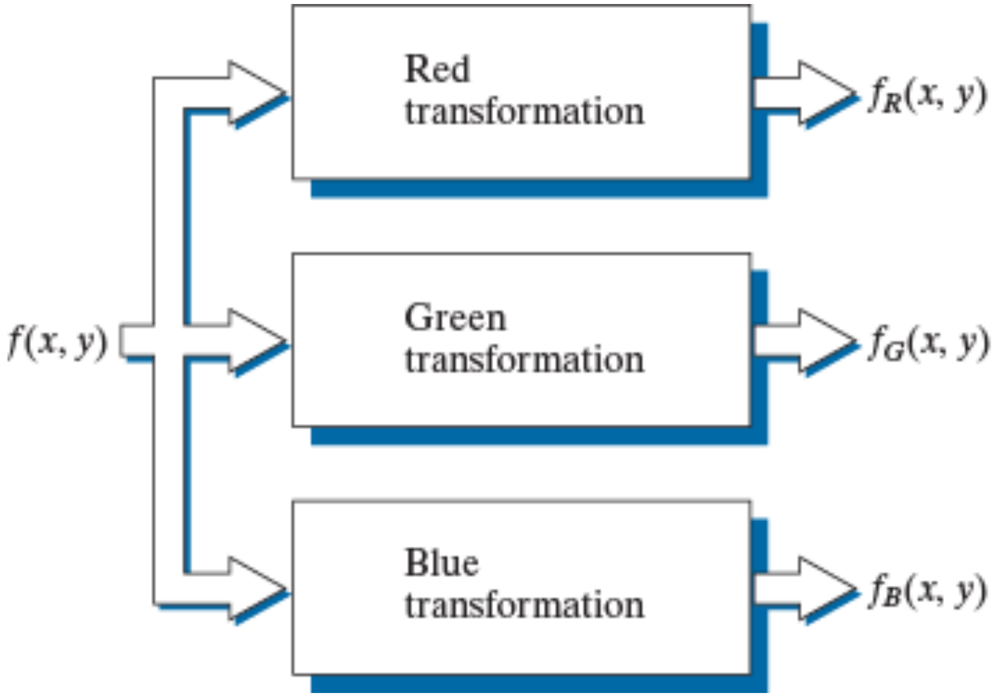
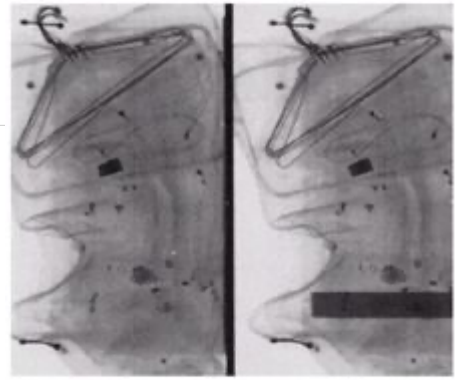
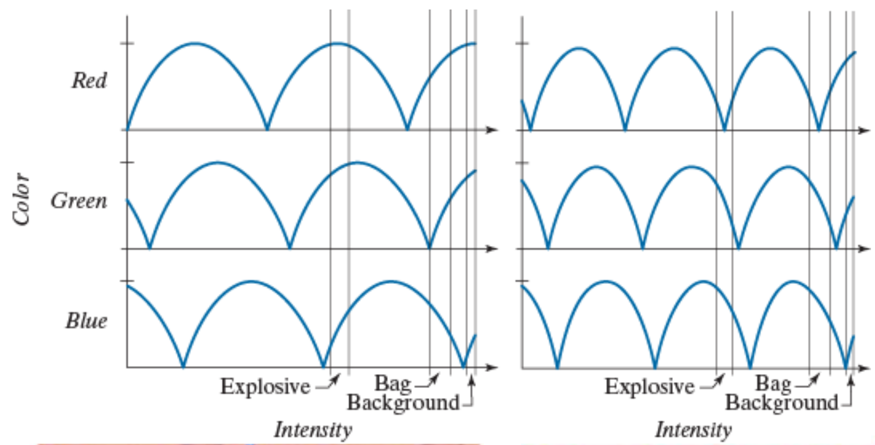


FIGURE 7.21 Functional block diagram for pseudocolor image processing. Images f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

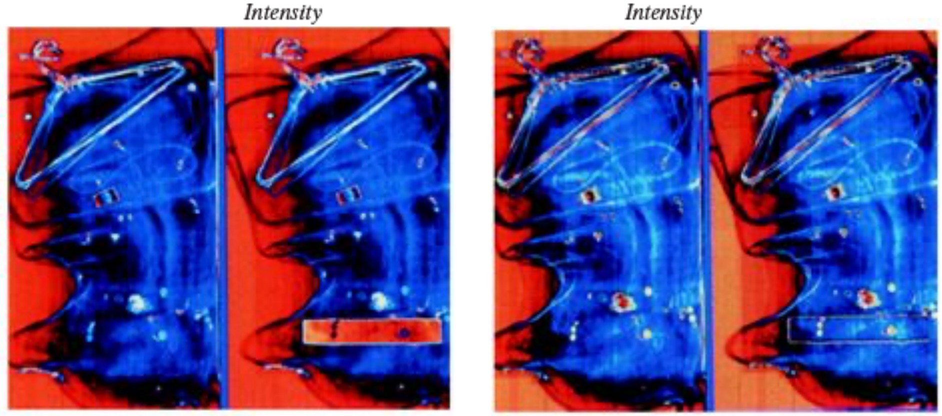
Match the transformation and the pseudocolor image

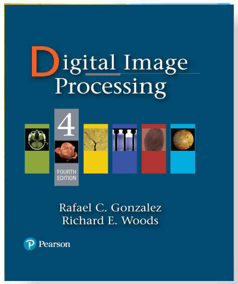




a b

FIGURE 7.23 Transformation functions used to obtain the pseudocolor images in Fig. 7.22.





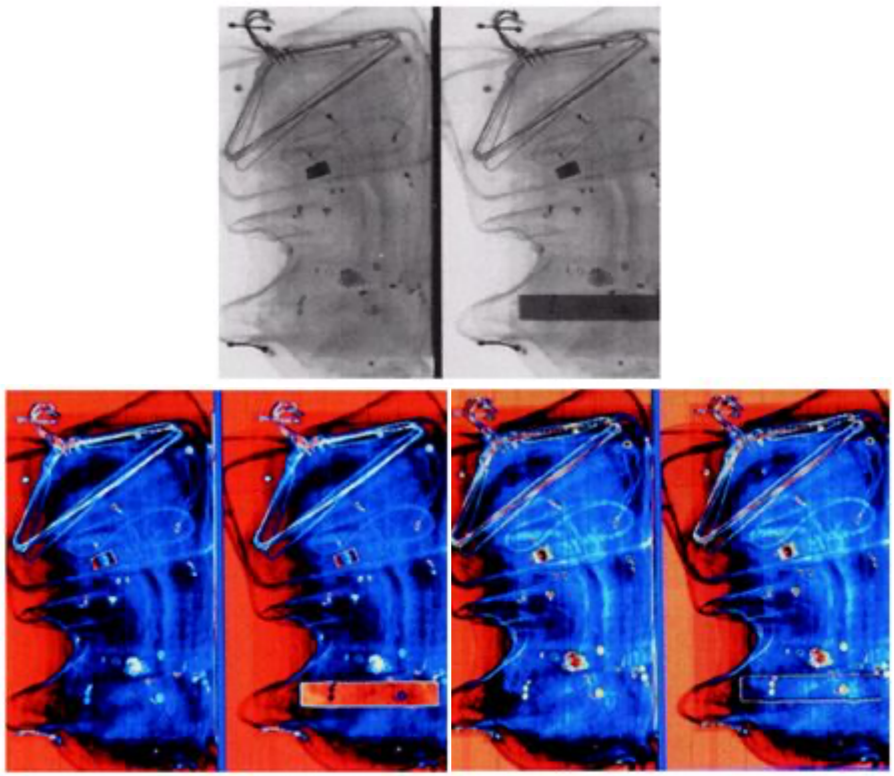
Chapter 7

Color Image Processing

a
b c

FIGURE 7.22 Pseudocolor enhancement by using the gray level to color transformations in Fig. 7.23. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)

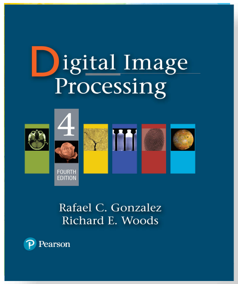
(a) Original X-ray



(b) (c)

(b) Good color separation for background, garment bag and explosives; however, background and explosives are assigned the same color due to the transformation used.

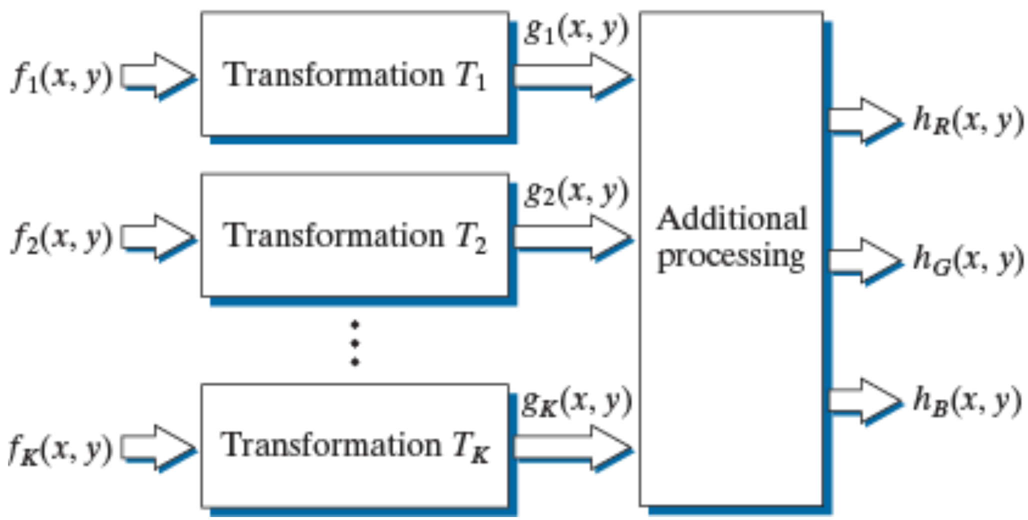
(c) Explosives and garment bag have nearly the same color, an observer can see through the hidden explosives!



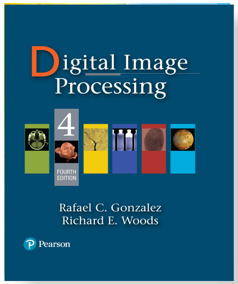
Pseudocoloring for Multispectral Images

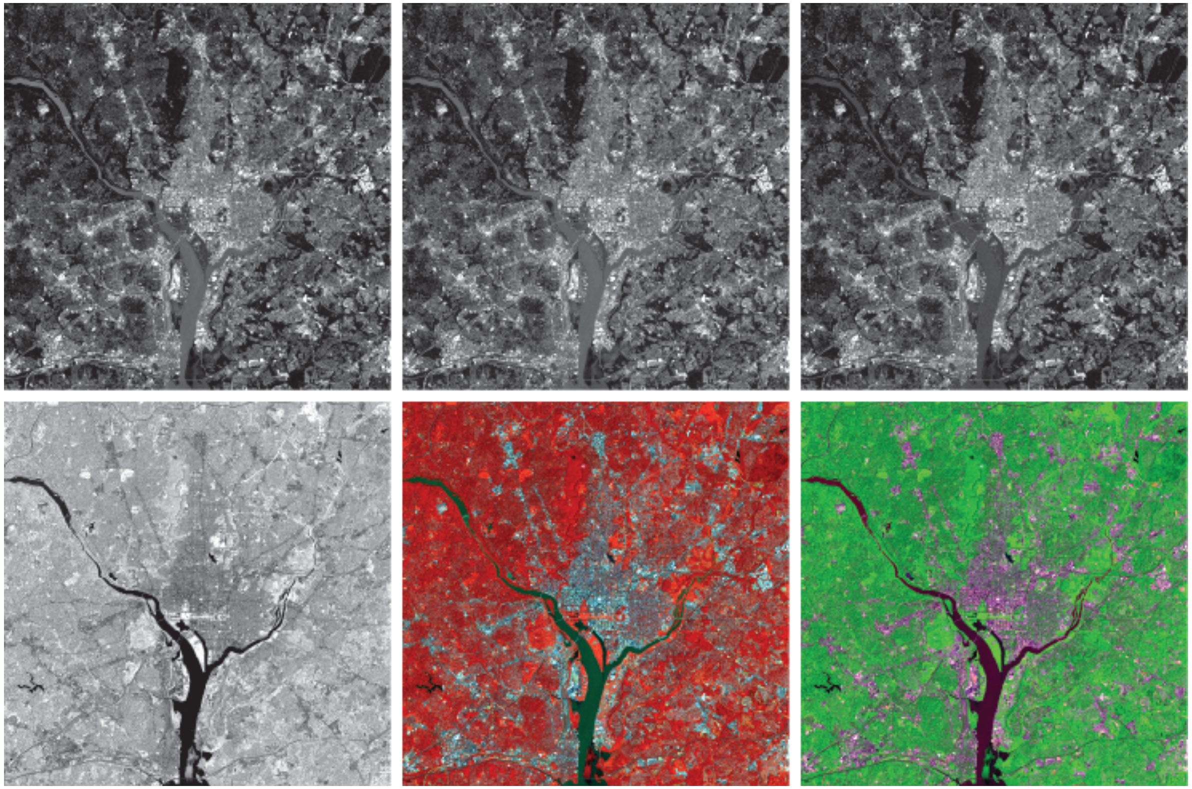
- Multispectral images have been acquired by different sensors at different wavelengths.
- Combining them to obtain a color image can be achieved as follows:

FIGURE 7.24 A pseudocolor coding approach using multiple grayscale images. The inputs are grayscale images. The outputs are the three components of an RGB composite image.



Additional processing may include color balancing, combining images, and selecting three of them for display.

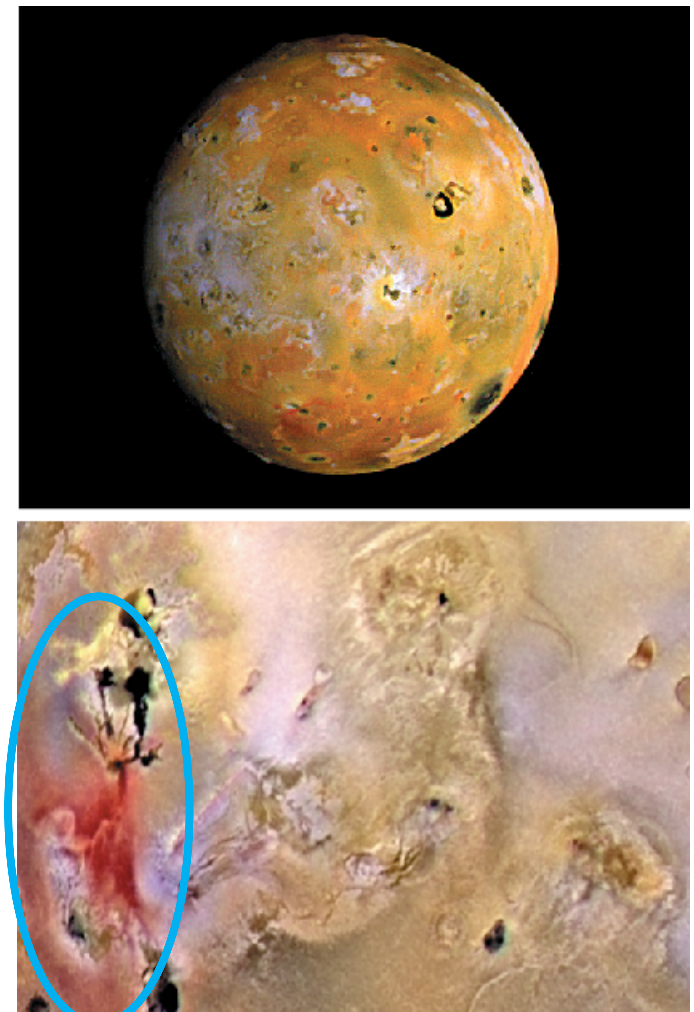




a b c
d e f

FIGURE 7.25 (a)–(d) Red (R), green (G), blue (B), and near-infrared (IR) components of a LANDSAT multispectral image of the Washington, D.C. area. (e) RGB color composite image obtained using the IR, G, and B component images. (f) RGB color composite image obtained using the R, IR, and B component images. (Original multispectral images courtesy of NASA.)

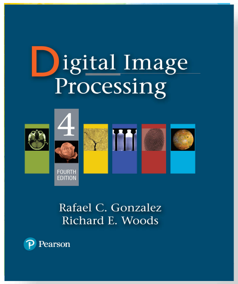
Pseudocoloring for Multispectral Images



a
b

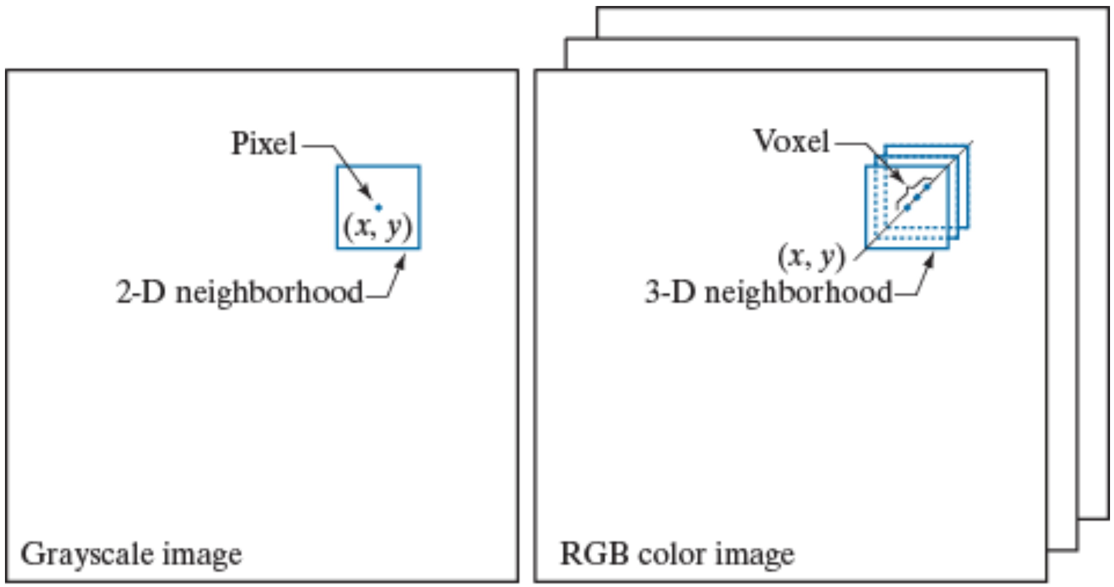
FIGURE 7.26 (a) Pseudocolor rendition of Jupiter Moon Io. (b) A close-up. (Courtesy of NASA.)

Regions of recent volcano activity



a b

FIGURE 7.27 Spatial neighborhoods for grayscale and RGB color images. Observe in (b) that a *single* pair of spatial coordinates, (x, y), addresses the same spatial location in all three images.



a color image is multi-valued, i.e. in RGB, each pixel has 3 values

Consider the following color transformation:

$$g(x, y) = T[f(x, y)]$$

Let r_i and s_i denote the color components of f and g, respectively,

$$s_i = T_i(r_1, r_2, \ldots, r_n) \quad \textit{where } n = 3 \textit{ or } 4$$

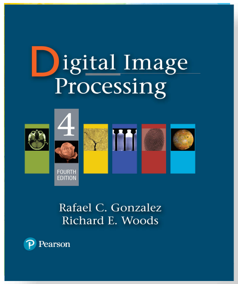
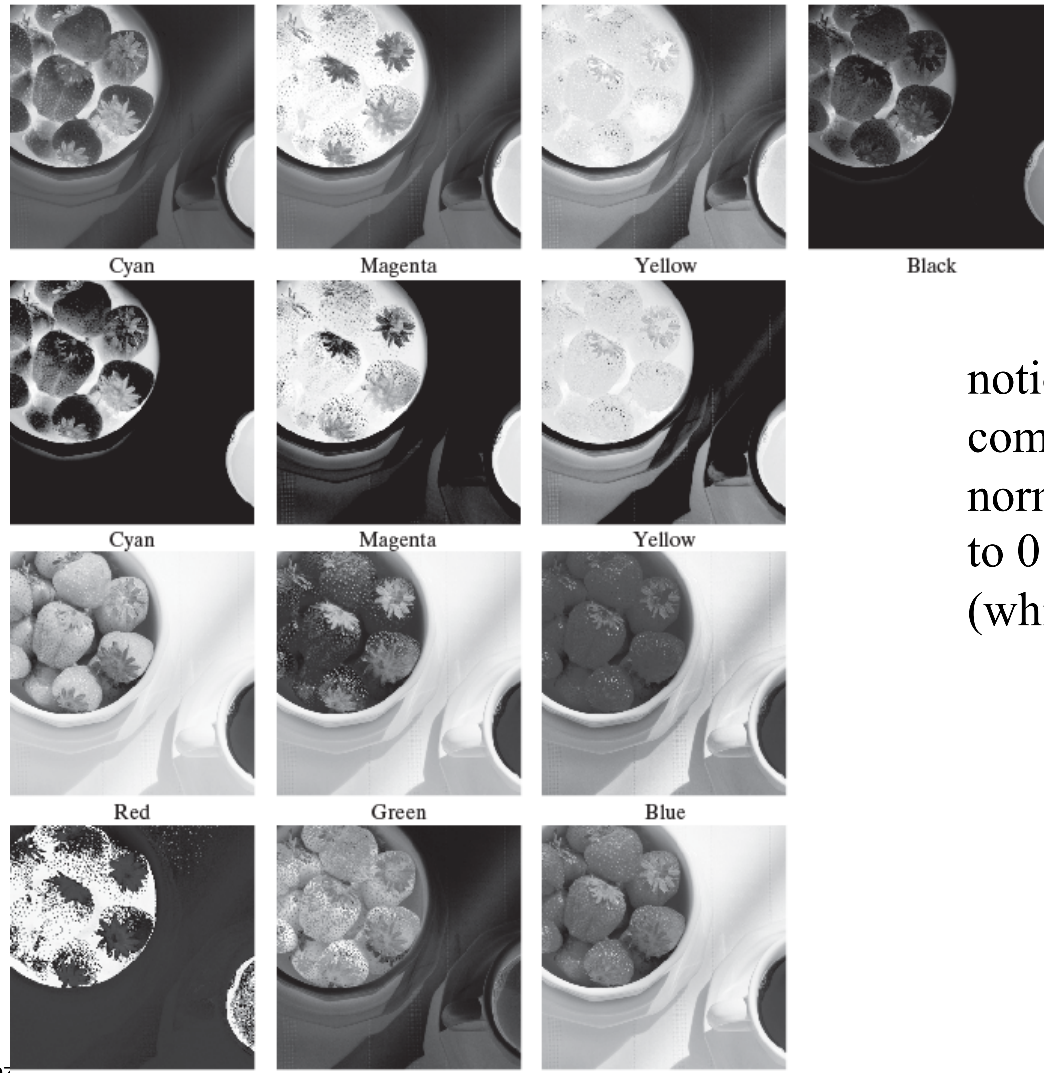


FIGURE 7.28 A full-color image and its various color-space components. (Original image courtesy of MedData Interactive.)

Full-Color Image Processing

Example of a full color and its color-space components

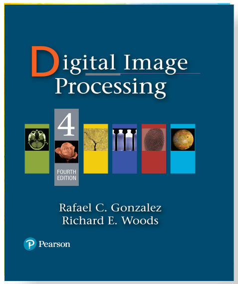
Full color image



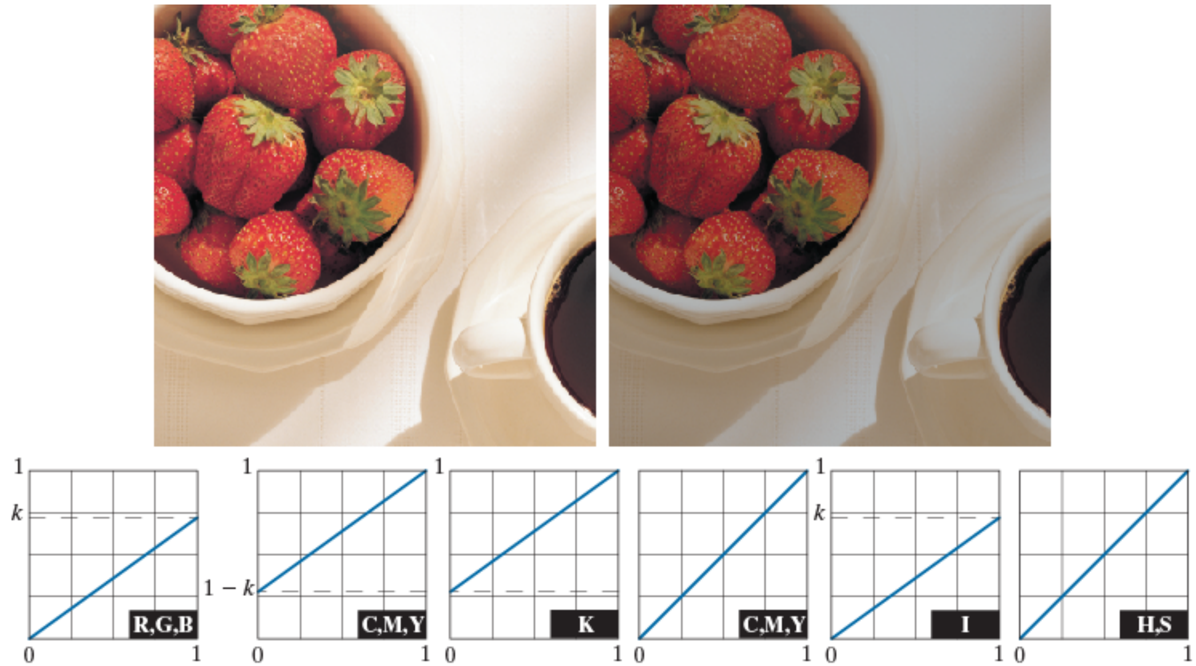
notice that all components are normalized to 0 (black) and 1 (white).

Color image component manipulation

How can you decrease the image intensity by 30%?



Color image component manipulation: adjusting the intensity component



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

c d e f g h

FIGURE 7.29 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) The required RGB mapping function. (d)–(e) The required CMYK mapping functions. (f) The required CMY mapping function. (g)–(h) The required HSI mapping functions. (Original image courtesy of MedData Interactive.)

Color Complements

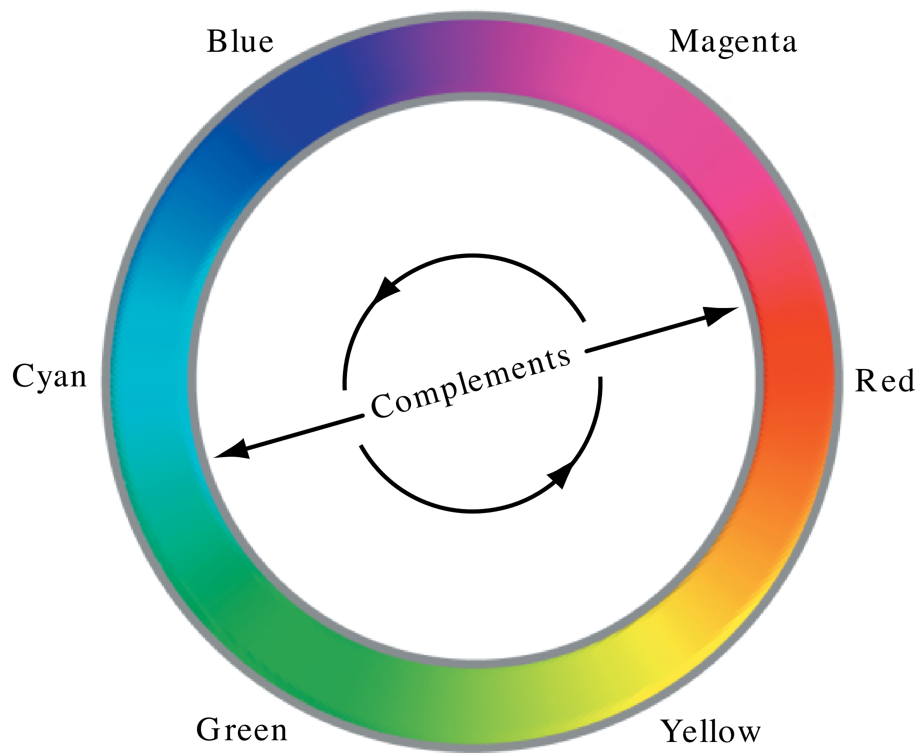
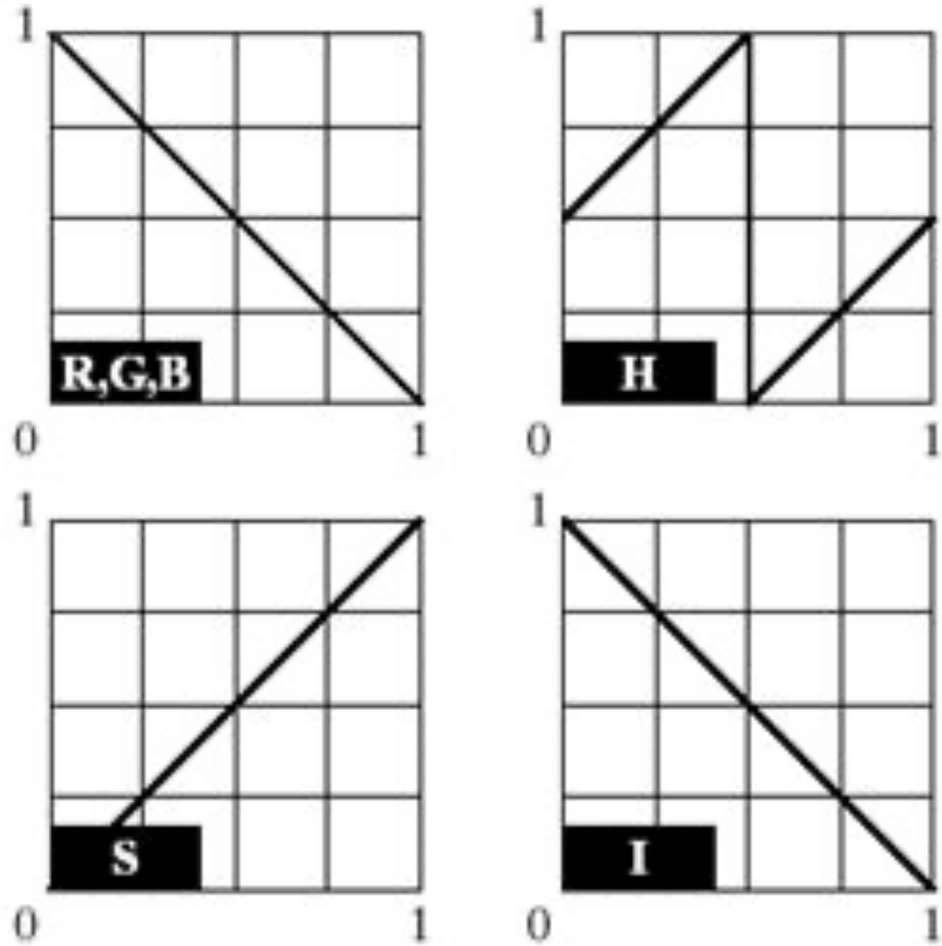
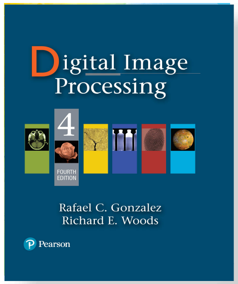


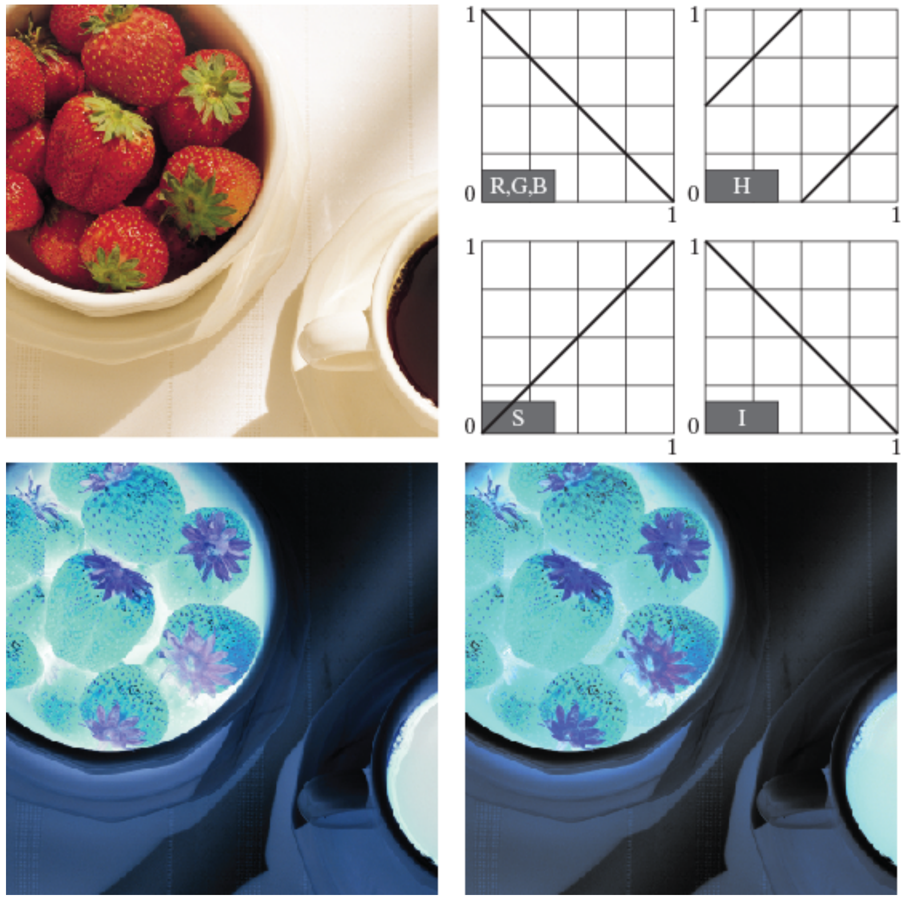
FIGURE 7.30 Color complements on the color circle.

What will these transformations do?





Color complement transformations



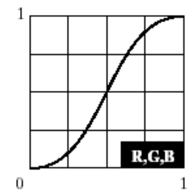
a b
c d

FIGURE 7.31 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.

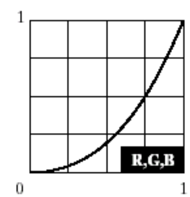
analogous to digital negatives for monochrome images

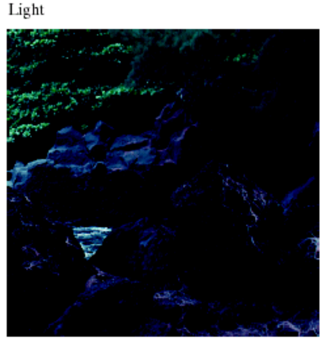
What will the transformations on the right do to the images on the left?

Flat

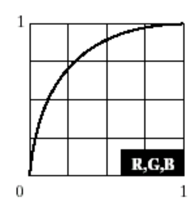


Light

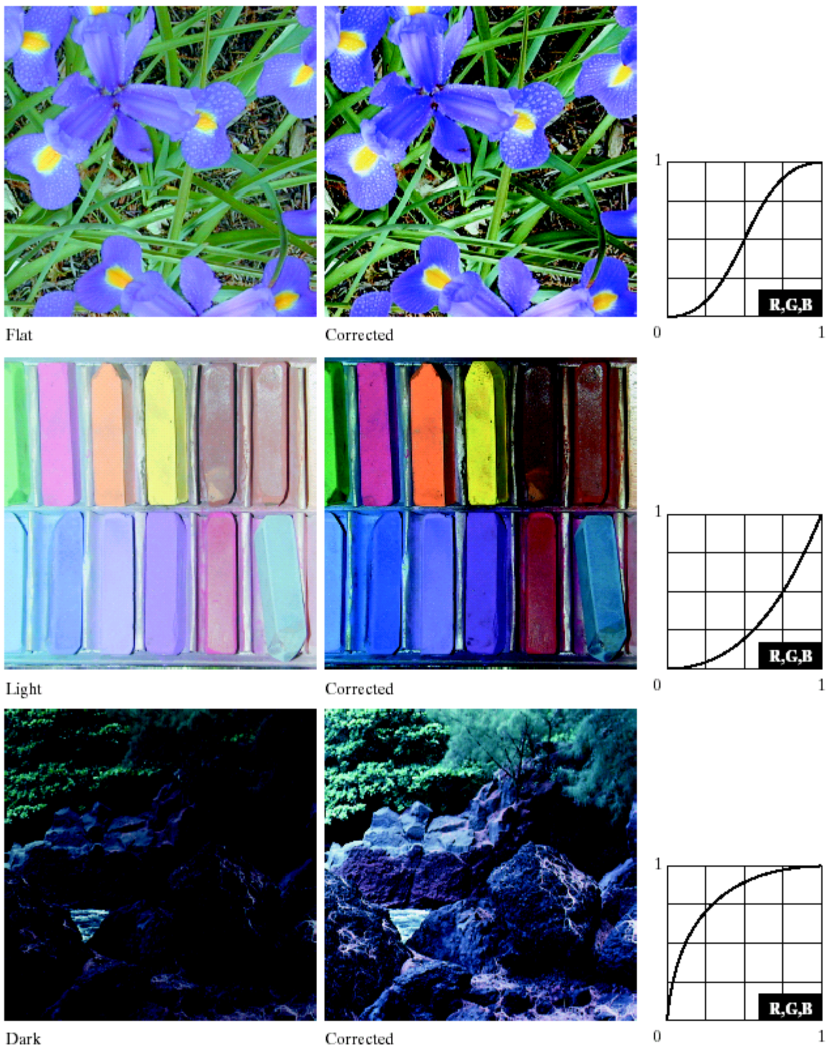




Dark



Color image histogram equalization for tonal correction



S-shaped transformation for boosting contrast

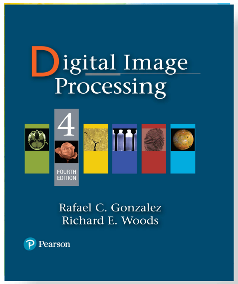
Power-law-like transformation to correct light and dark details, as in grey-scale images.

Equalizing the color components equally does not always change the image hues significantly.

FIGURE 7.33 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.

Tone and Color Corrections/Balancing

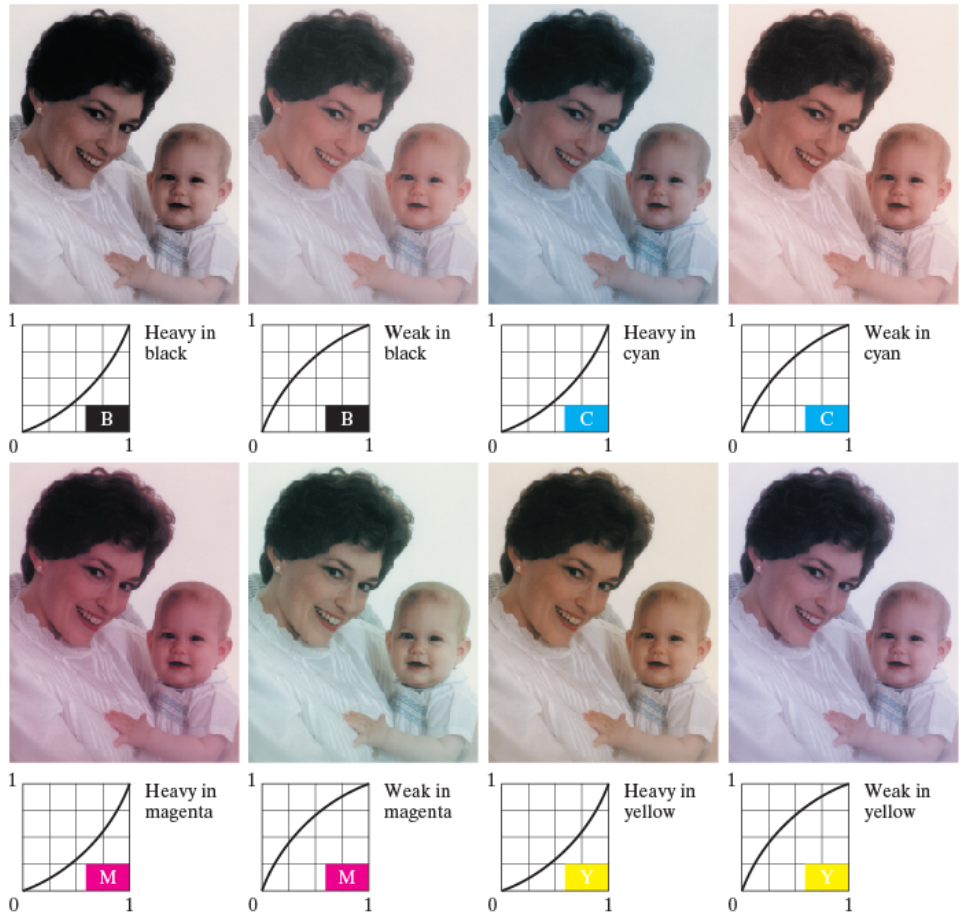
- Goal: correct color image through pixel transformations to get a better visualization and/or print out.
- Lab color space is perceptually uniform, i.e. color differences are perceived uniformly.
- Like HSI, Lab decouples intensity from color
- Example: tonal correction for three common tonal imbalances: flat, light and dark images, see previous slide.



Color Balancing

Original/Corrected

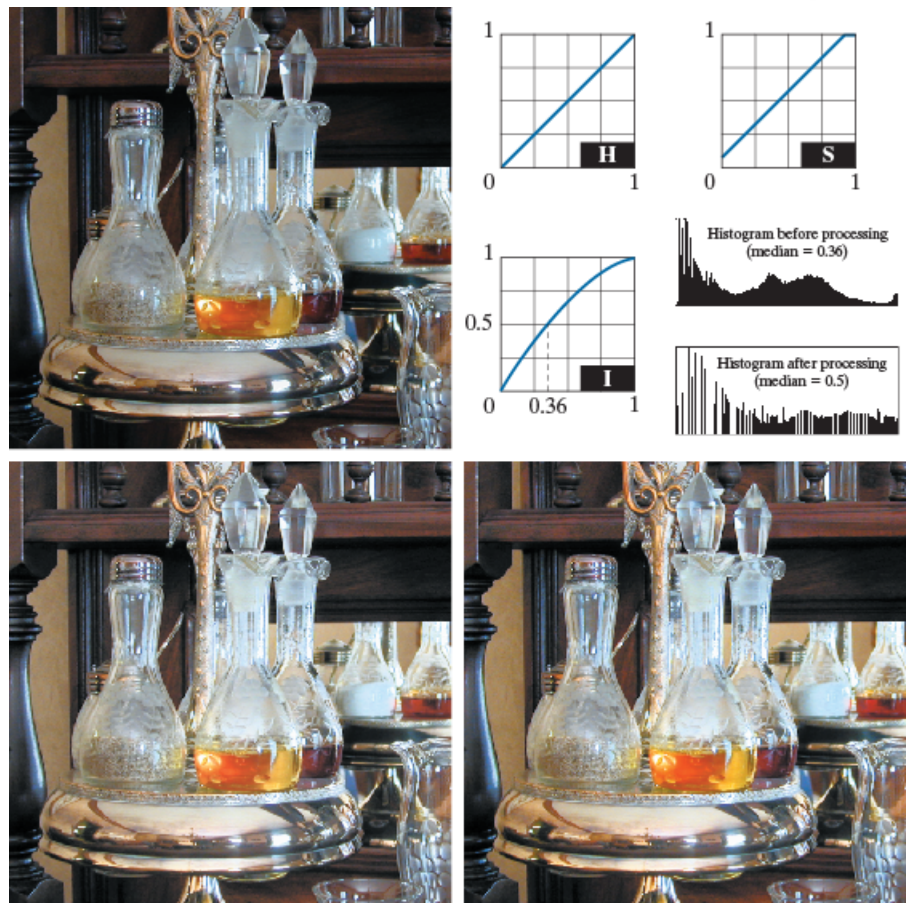
FIGURE 7.34 Color balancing a CMYK image.



Histogram Equalization for Color Images

- Applying a histogram equalization on all image components might change the image colors!
- Which color space or model would you use?
- Which component(s) would you equalize?

Histogram Equalization for Color Images



a b
c d

FIGURE 7.35 Histogram equalization (followed by saturation adjustment) in the HSI color space.

(a) original

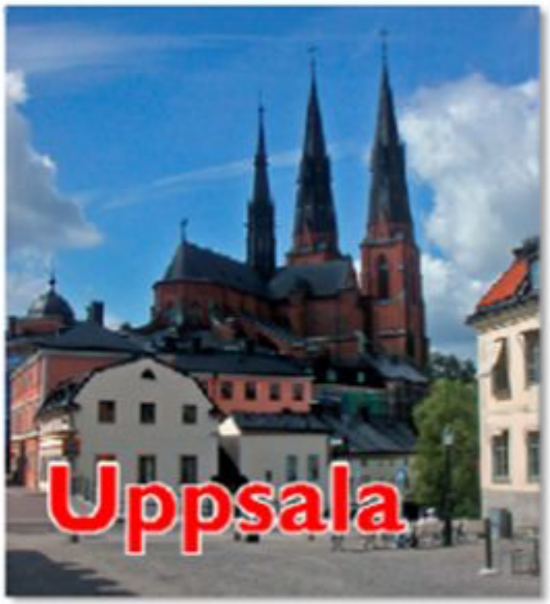
(c) Equalized image by modifying only I.

(d) Equalized image by modifying I and slightly S.

Two remarks:

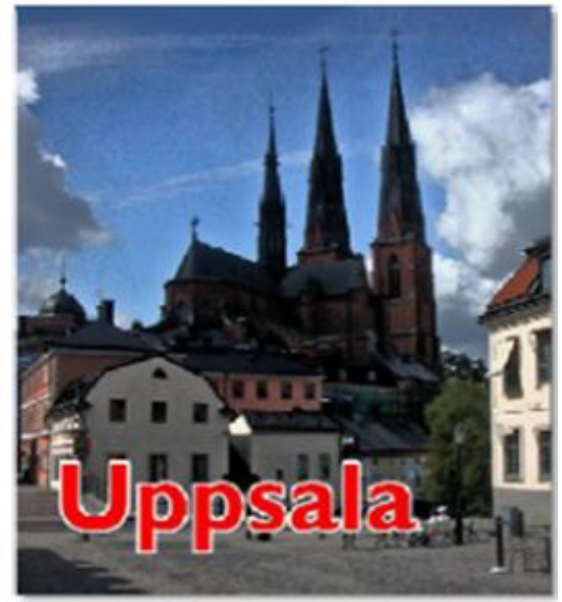
1) Notice the effect on the histogram,
2) Notice the liquid color, it’s not as crisp as in the original.

Histogram Equalization for Color Images



Original image

RGB image
histogram equalized for each channel individually



HSI image
histogram equalized for the I-channel only

Histogram Equalization for Color Images

Question:
Would it be wise to equalize RGB color components?

Answer:
No, this way the color will change

Recommendation:
Equalize intensity component only in HSI space

Color Image Processing

a	b
c	d

FIGURE 7.36
(a) RGB image.
(b) Red component image.
(c)Green component.
(d) Blue component.

a b c

FIGURE 7.37 HSI components of the RGB color image in Fig. 7.36(a). (a) Hue. (b) Saturation. (c) Intensity.

RGB Image Smoothing and Sharpening

Local Average

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

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

Laplacian

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

Color Image Smoothing

Smoothing all color components versus smoothing only intensity component:

a b c

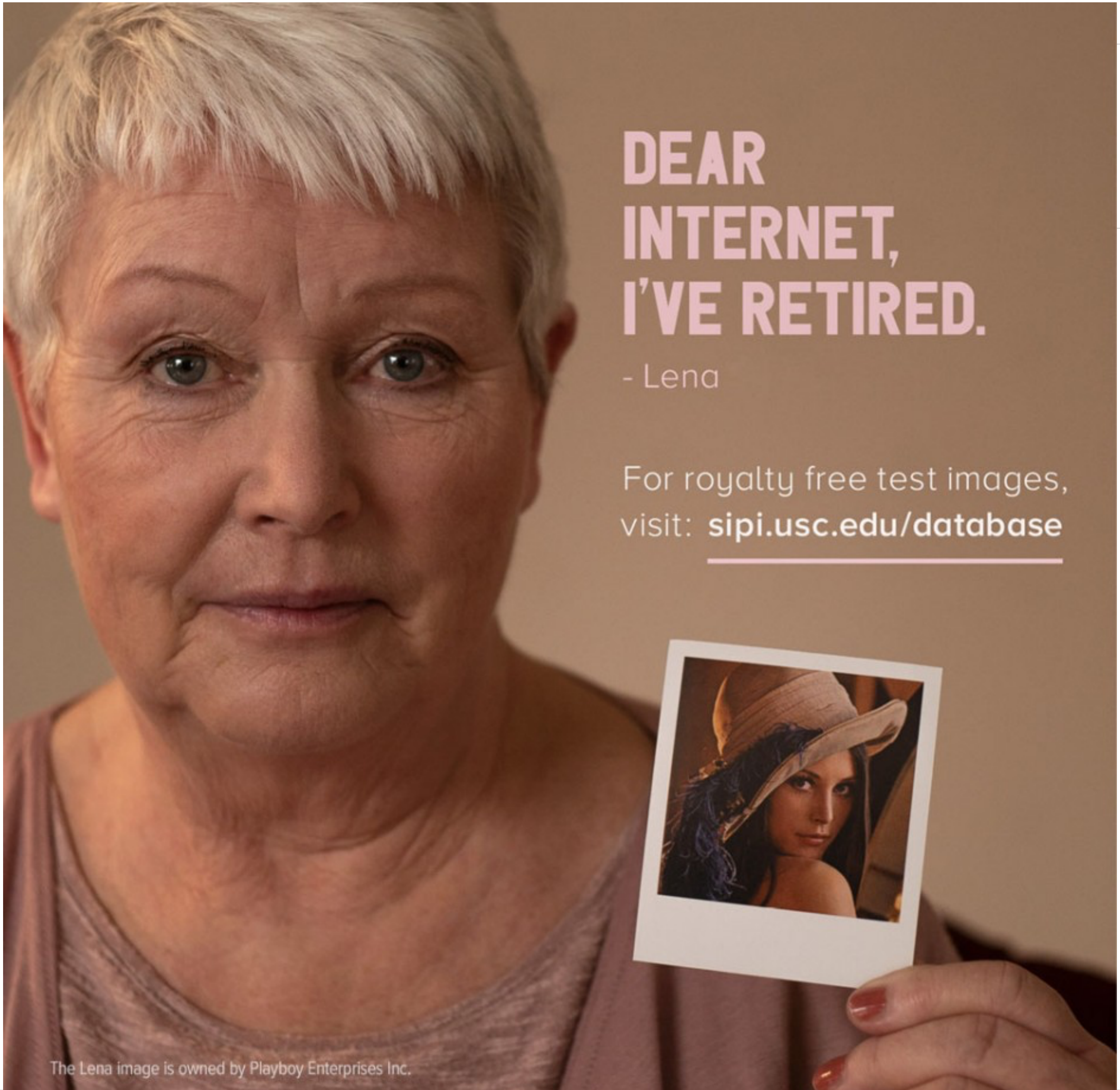
FIGURE 7.38 Image smoothing with a 5×5 averaging kernel. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

Note that (a) is smoother and lost some of its original colors, in contrast to (b) which preserved its hue and saturation.

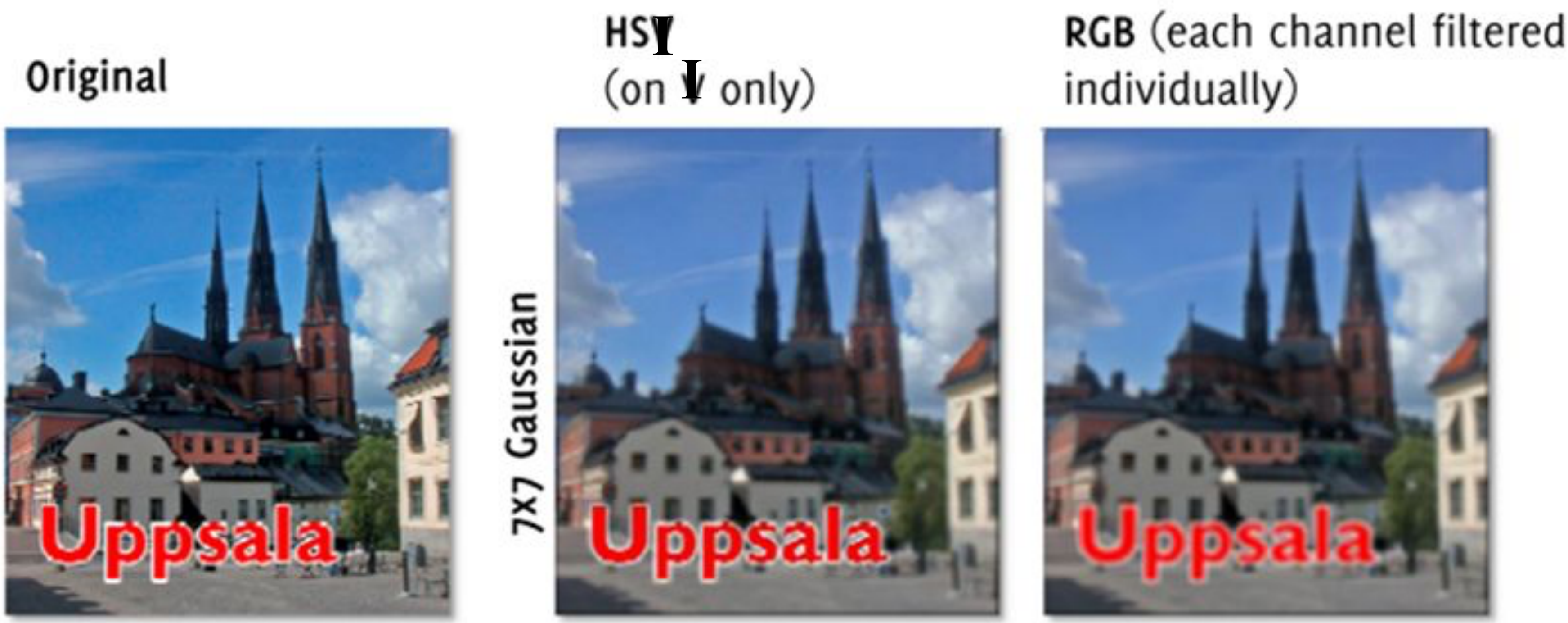
Color Image Sharpening

a b c

FIGURE 7.39 Image sharpening using 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.


DEAR
INTERNET,
I'VE RETIRED.
- Lena
For royalty free test images,
visit: sipi.usc.edu/database
The Lena image is owned by Playboy Enterprises Inc.

Color Image Smoothing



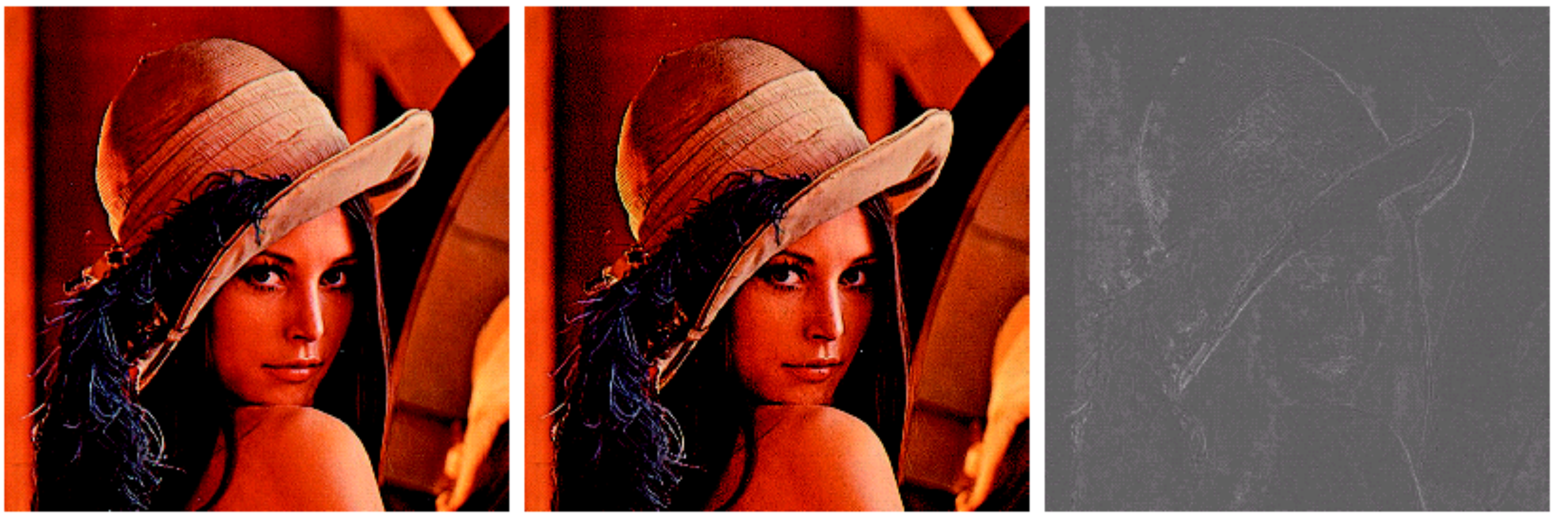
Color Image Smoothing

- You might end up with **color artefacts** if H-channel is filtered
- Remember: H-channel consists of angles 0-360°

H-channel blurred with a 3x3 Gaussian

Color Image Sharpening

Sharpening all color components versus sharpening only intensity component:



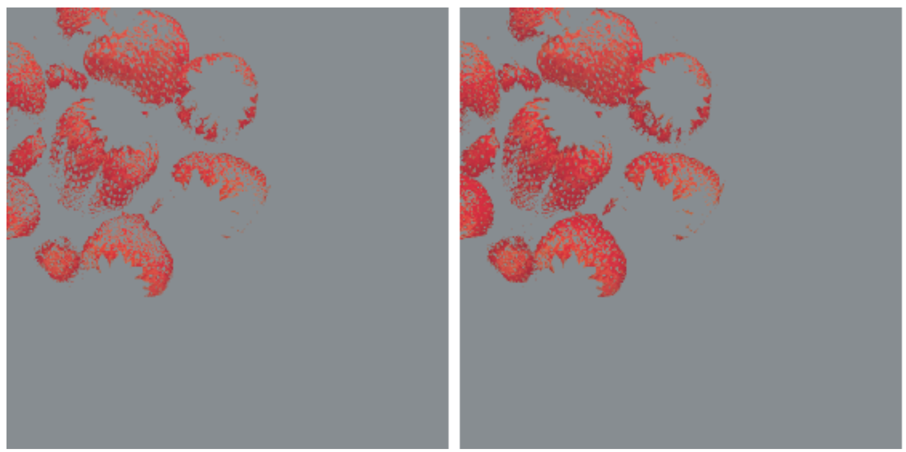
a b c

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.

Color Segmentation

How to keep red colors only and make everything else grey?

Remember color slicing!



a b

FIGURE 7.32 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 $(0.5, 0.5, 0.5)$.

How to apply color slicing?

Idea: highlight a range of colors in an image in order to

- separate them from background, or
- use the region defined by color mask for further processing, e.g. segmentation

This is a complex extension of gray level slicing due to the multi-valued nature of color images

How can this be done? Can map the colors outside some range of interest to some fixed color and leave the rest as they are. Let $\boldsymbol{a}=(a_1,a_2,a_3)$ be the average of the color region of interest and W the width of this region, then

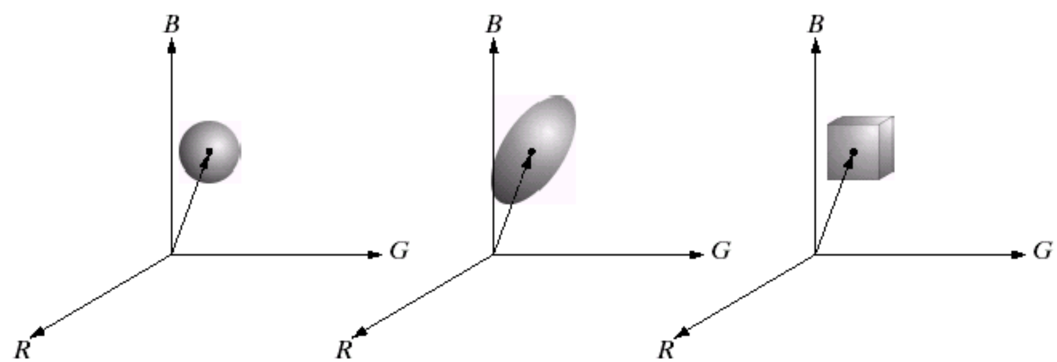
$$s_i = \begin{cases} 0{,}5 & if \left[|r_j - a_j| > \frac{W}{2} \right]_{\forall 1 \le j \le 3} \\ r_i & \end{cases} \qquad \text{for } i=1,2,3$$

If a sphere is used to specify the region of interest, then

$$s_i = \begin{cases} 0{,}5 & if \sum_{j=1}^{3} (r_j - a_j)^2 \succ R_0^2 \\ r_i & \end{cases}$$

Segmentation in RGB Color Space
An extension of color slicing

Suppose that regions of specific color range are to be segmented. The specific color is specified by an average color **a** and a neighborhood around it, defined by a suitable distance measure: we say that **z** is similar to **a** if D(**a**,**z**) is smaller than a threshold D_0.



a b c

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

(a) Euclidean distance (most general)
(b) Malahanobis distance (take into account properties of the data)
(c) Bounding box (reduce computational complexity)

Segmentation in HSI Color Space

a b
c d
e f
g h

FIGURE 7.40 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 from (a).

Problem:
Segment the reddish region in the lower left side of the image.

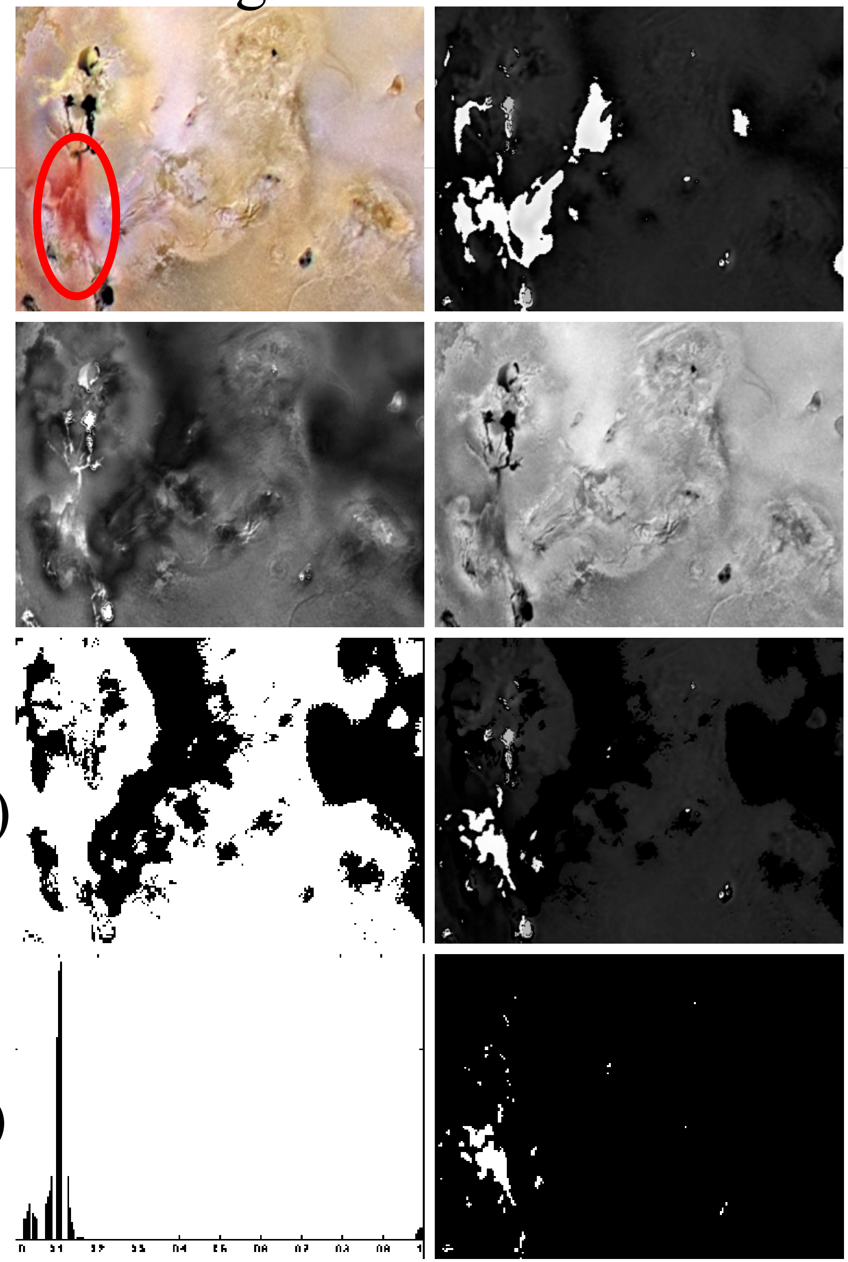
Proposed solution:
The region of interest has high **hue** values in the blue-magenta side of Red.
Let us prepare a mask by thresholding the saturation at 10% of the max value (e), and
Multiply the hue and thresholded saturation to enhance the areas of high saturation in the hue.
Then, we take the histogram of (f) and notice the high values of the reddish pixels near 1 (g).
Finally, segment in pixel values in (a) to obtain the result in (h).

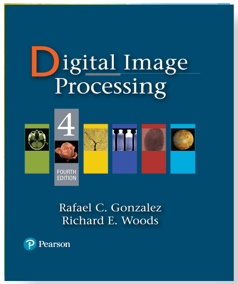
original hue

saturation intensity

(e) (f)

(g) (h)





Alternative solution

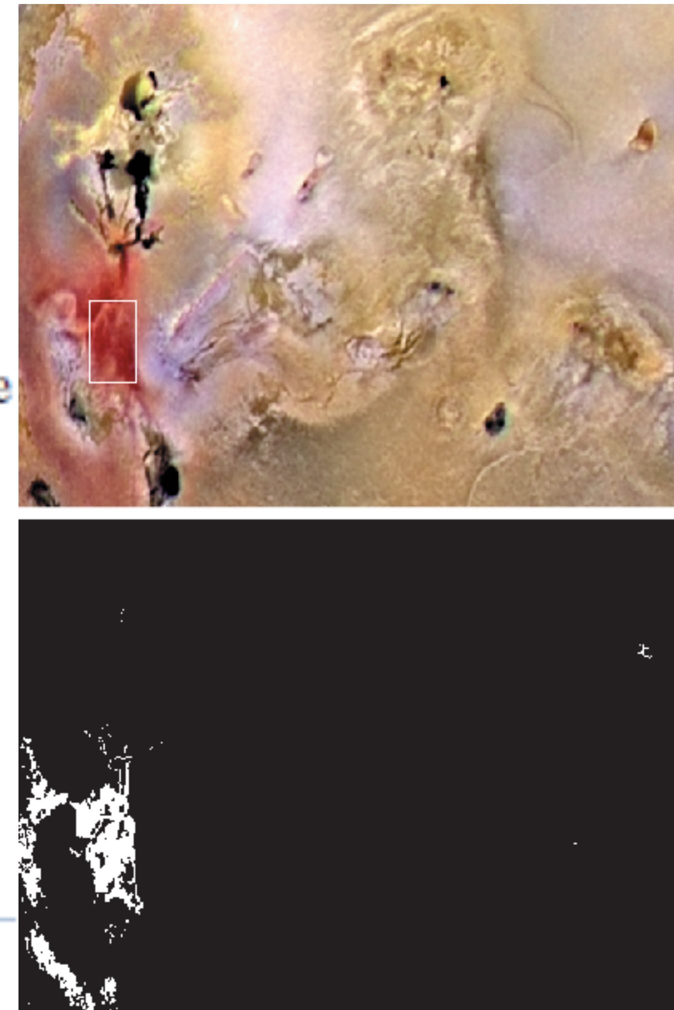
Want to segment reddish colors in the image, as in the rectangular region.

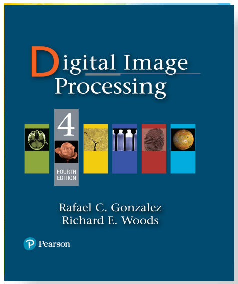
1. Compute the mean vector *a*
2. Use a bounding box of size
   2.5 x std deviation in each color
3. Color as black any pixel whose colors fall outside the box and as white the other pixels.

This segmentation is more accurate compared to the one performed in HSI, see Fig. 7.40 (h) (previous slide)

a
b

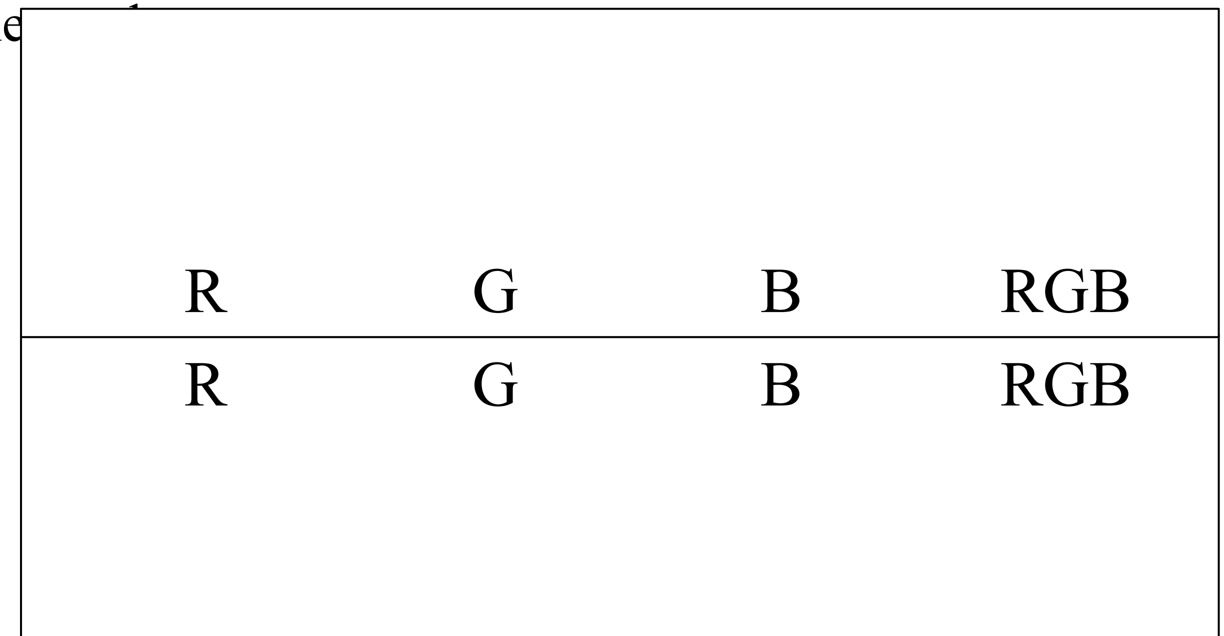
FIGURE 7.42 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 7.40(h).





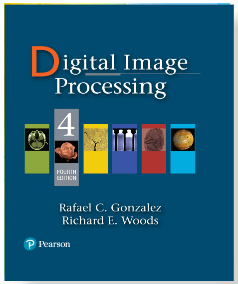
Computing image gradients

- We used the gradient operator to detect edges and details in grey level images.
- How can we use gradients to locate edges in a color (multi-component) image?
- Let us see what happens when we apply gradient to each component and then combine the

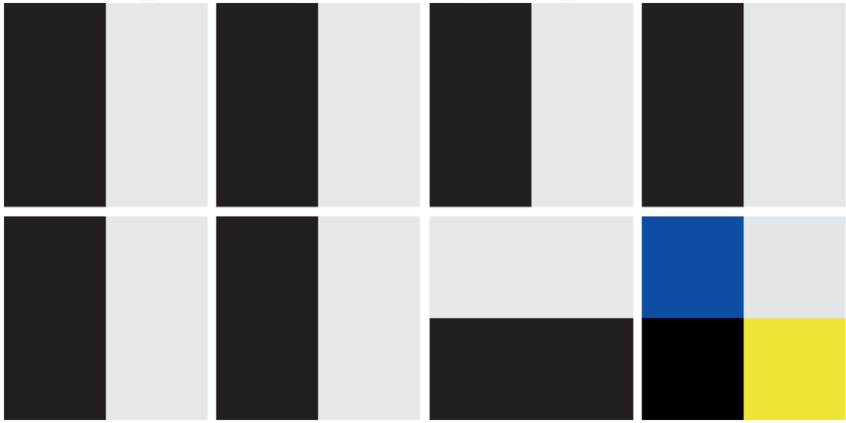


a b c d
e f g h

FIGURE 7.43 (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.



Computing image gradients



Conclusions:

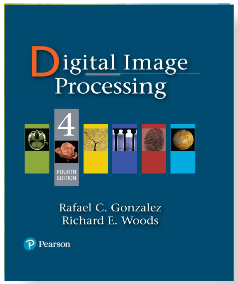
- Component-wise gradient can locate edges in a color image, but it is not accurate.
- If accuracy is not an issue, we can use locate edges in a color image using scalar gradients.
- More accurate gradient (vector pointing in the direction of maximum rate of change of f at a given point (x,y)) can be obtained via a proper definition of the gradient for vector quantities.

a b
c d

FIGURE 7.44
(a) RGB image. (b) Gradient computed in RGB color vector space. (c) Gradient image formed by the elementwise sum of three individual gradient images, each computed using the Sobel operators. (d) Difference between (b) and (c).

a b c

FIGURE 7.45 Component gradient images of the color image in Fig. 7.44. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 7.44(c).



Noise Filtering in Color Images

a	b
c	d

FIGURE 7.46 (a)–(c) Red, green, and blue 8-bit component images corrupted by additive Gaussian noise of mean 0 and standard deviation of 28 intensity levels. (d) Resulting RGB image. [Compare (d) with Fig. 7.44(a).]

Consider the RGB components, each was corrupted with Gaussian noise (0,28).

None of the components looks very objectionable including the color image!

Noise Filtering in Color Images

Now, convert the same image to HSI and look at the components!

a b c

FIGURE 7.47 HSI components of the noisy color image in Fig. 7.46(d). (a) Hue. (b) Saturation. (c) Intensity.

This is due to the nonlinearities in the conversion between RGB and HSI. The intensity component I does not look bad, why?

Noise Filtering in Color Images

a b
c d

FIGURE 7.48
(a) RGB image with green plane corrupted by salt-and-pepper noise.
(b) Hue component of HSI image.
(c) Saturation component.
(d) Intensity component.

Suppose that only one of the RGB channel is corrupted with noise, say the green channel, converting this to HSI yields a high degradation in the hue and saturation components.