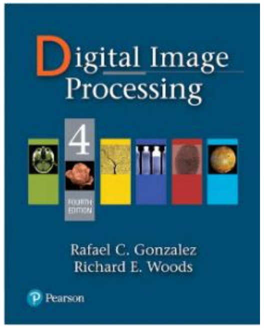


Digital Image Processing

Color Image Processing (II)

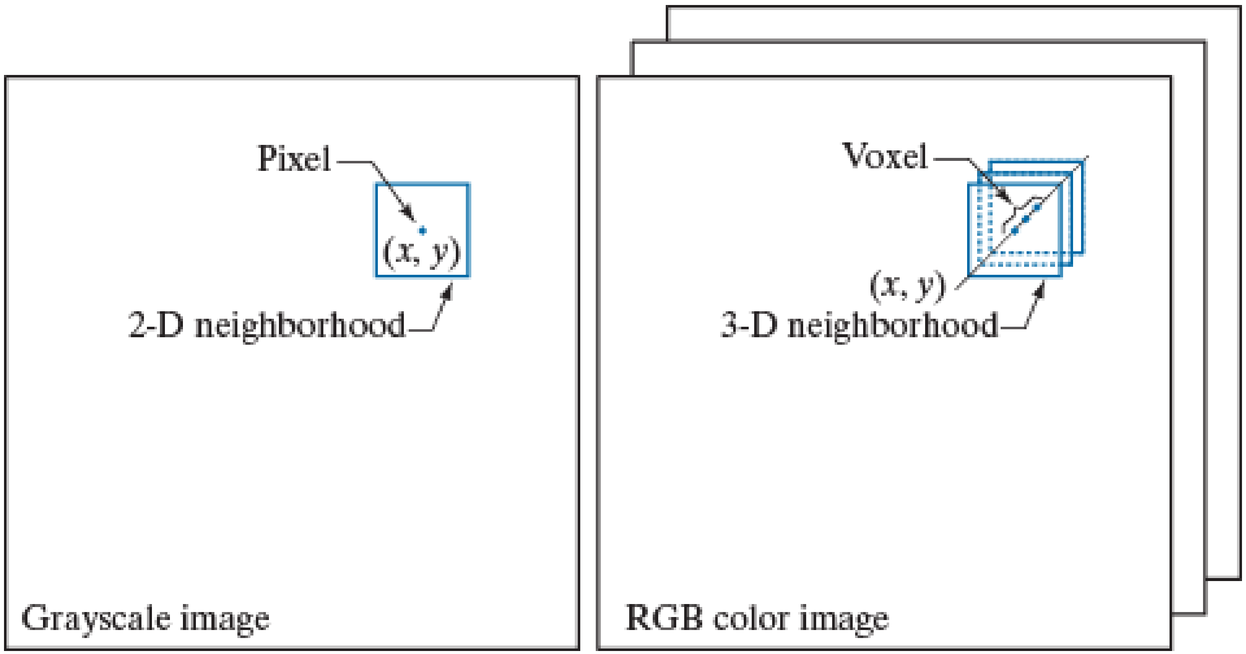
Dr. Tun-Wen Pai

1) **Color transformation (Intensity modification)**
2) **Color complementary**
3) **Color histogram equalization**
4) **Color smoothing/sharpening/noise filtering**

Dec. 12, 2022

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.



Color Transformations

- It is useful to think of a color image as vector valued image, where each **pixel has associated with it, as vector of three values**.

- Each components of this vector corresponds to different aspect of color, depending on the color model being used. For example, in an RGB model, the three values in the vector respectively denote the red, green, and blue components of the color of that pixel. In an HSI model, the three values in the vector denote the hue, saturation, and intensity of the color of that pixel.

- We can think of **color transformations** as a transformation of vectors.

$$s_i = T_i(r1, r2, r3), \quad \mathrm{i} = 1,2,3.$$

- Here (r1, r2, r3 represent the color components of the input image f(m, n), whereas (s_1, s_2, s_3) represent the color components of the output image g(m, n)).

- In theory, any color transformation can be performed in any color space model. However, in practice, some transformations are better suited to specific models.

- Moreover, the cost of conversion between the models must be a factor in implementations of a particular transformation.

Example: Modifying intensity

- Consider a simple transformation involving **intensity scaling**:

$$g(m,n) = kf(m,n)$$

$$\text{where } 0 < k < 1 \text{ is a scaling factor}$$

- In **HSI space**, this can be implemented as

$$s_1 = r_1, \quad s_2 = r_2, \quad \text{and} \quad s_3 = kr_3$$

- In **RGB space**, this can be implemented as

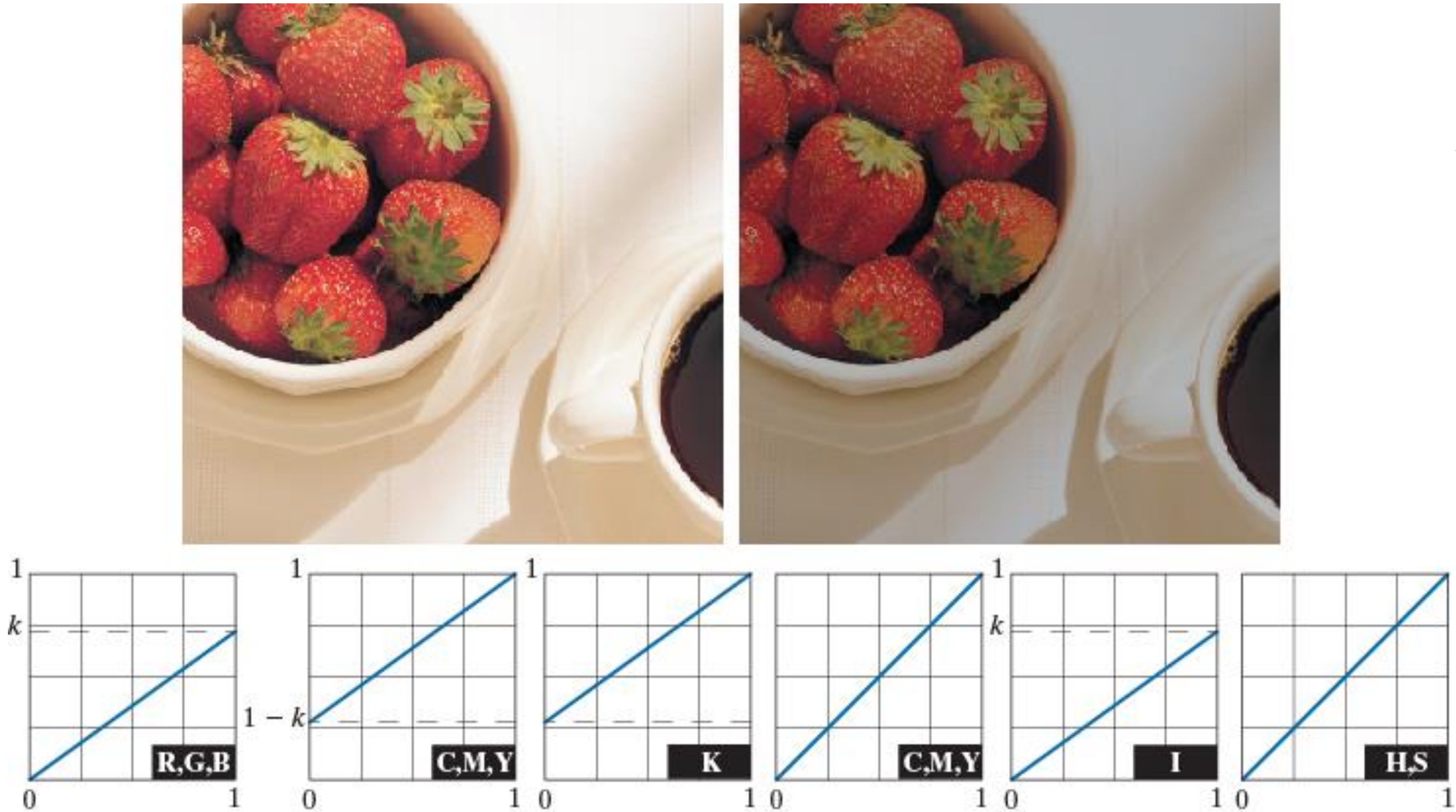
$$s_1 = kr_1, \quad s_2 = kr_2, \quad \text{and} \quad s_3 = kr_3$$

- In **CMY space**, this can be implemented as

$$s_1 = kr_1 + 1 - k, \quad s_2 = kr_2 + 1 - k, \quad \text{and} \quad s_3 = kr_3 + 1 - k$$

- Although the fewest operations are involved in the HSI space, the computations involved in conversion back and forth from RGB space more than offsets any savings in computation in HSI space.

FIGURE 6.28 A full-color image and its various color-space components. (Original image courtesy of Med-Data Interactive)



a 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 Complement

- Hues opposite one another in a color circle are called **complements.**

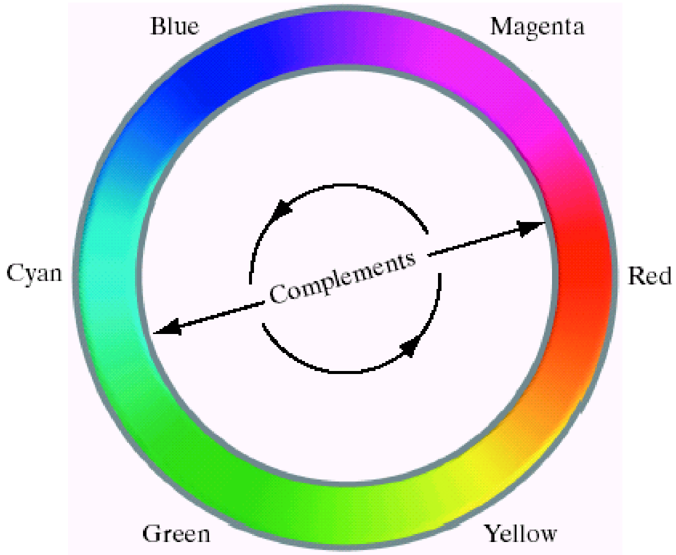
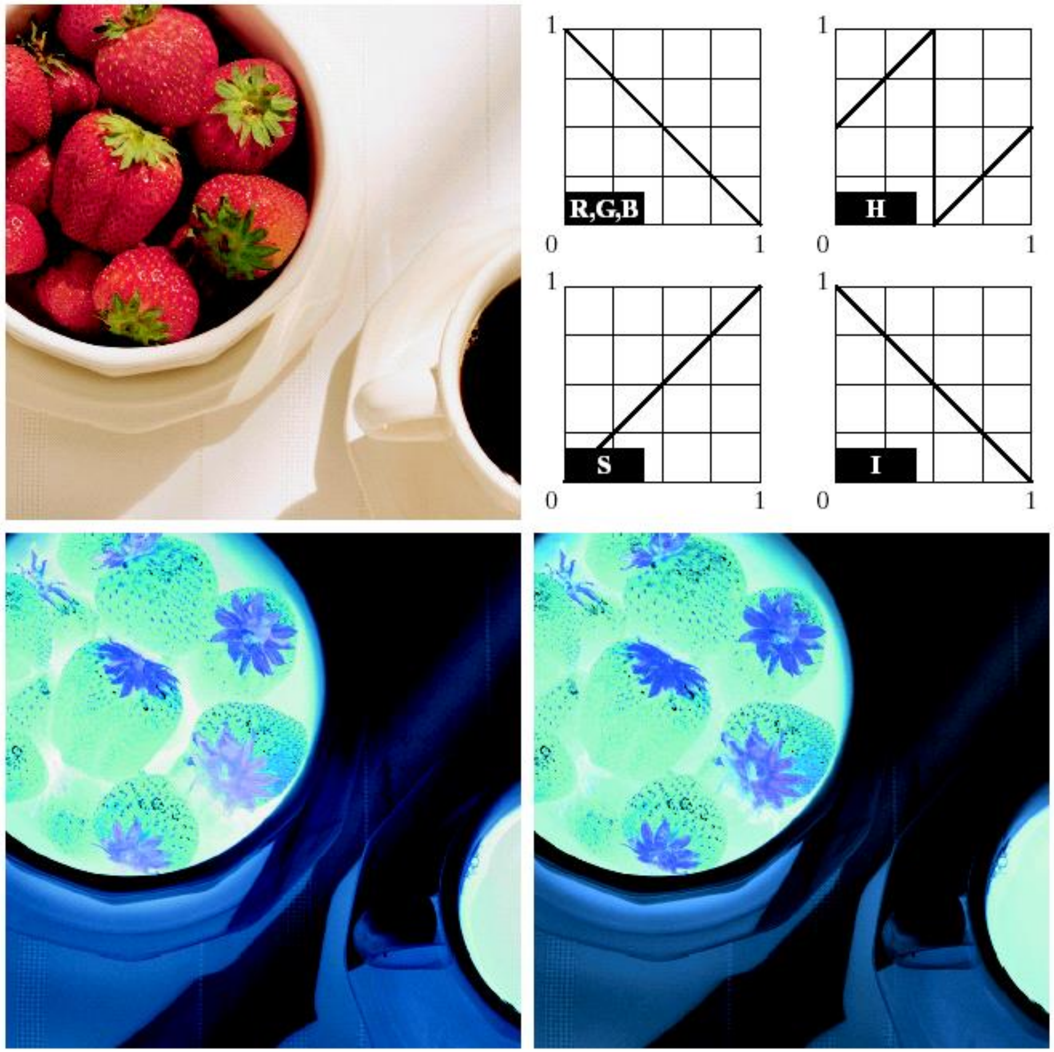


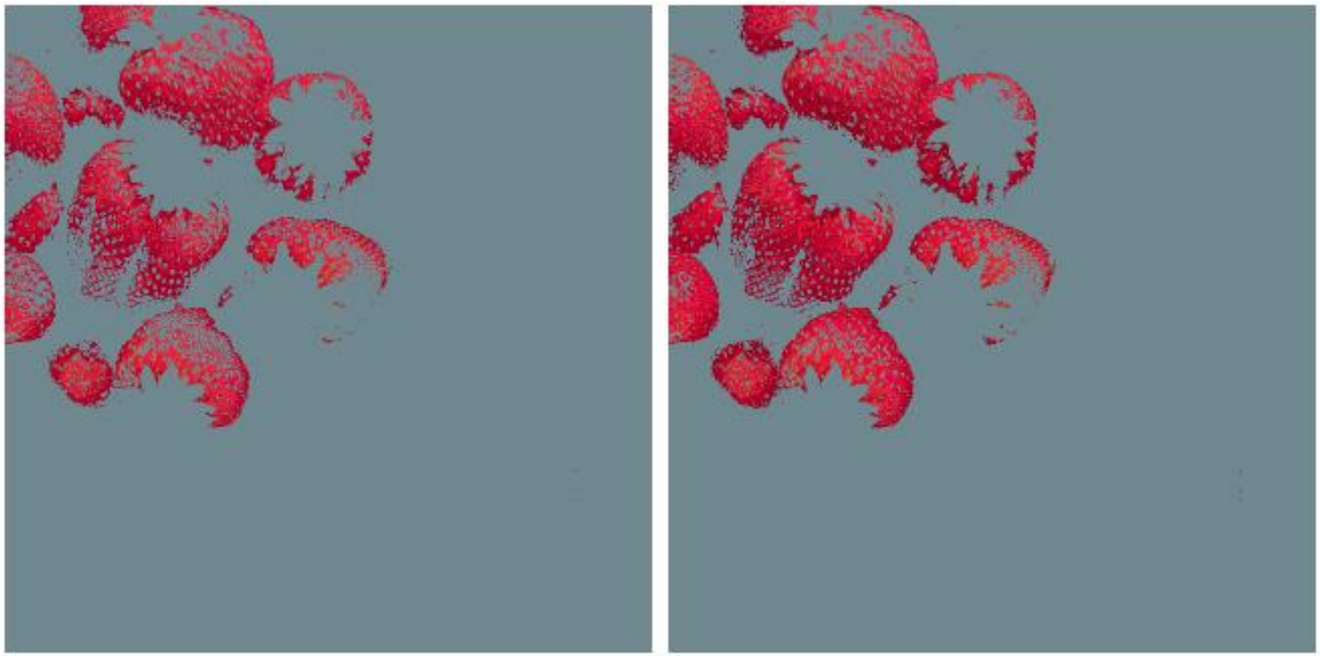
FIGURE 6.30
Complements on the color circle.

- This is analogous to gray-scale negatives.
- As in the grayscale case, this transformation is useful in enhancing details embedded in dark portions of a color image.
- Complementation can be easily implemented in the RGB space.
  However, there is no simple equivalent of this in the HIS space. An approximation is possible.



a b
c d

FIGURE 6.31 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.



a b

FIGURE 6.32 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).

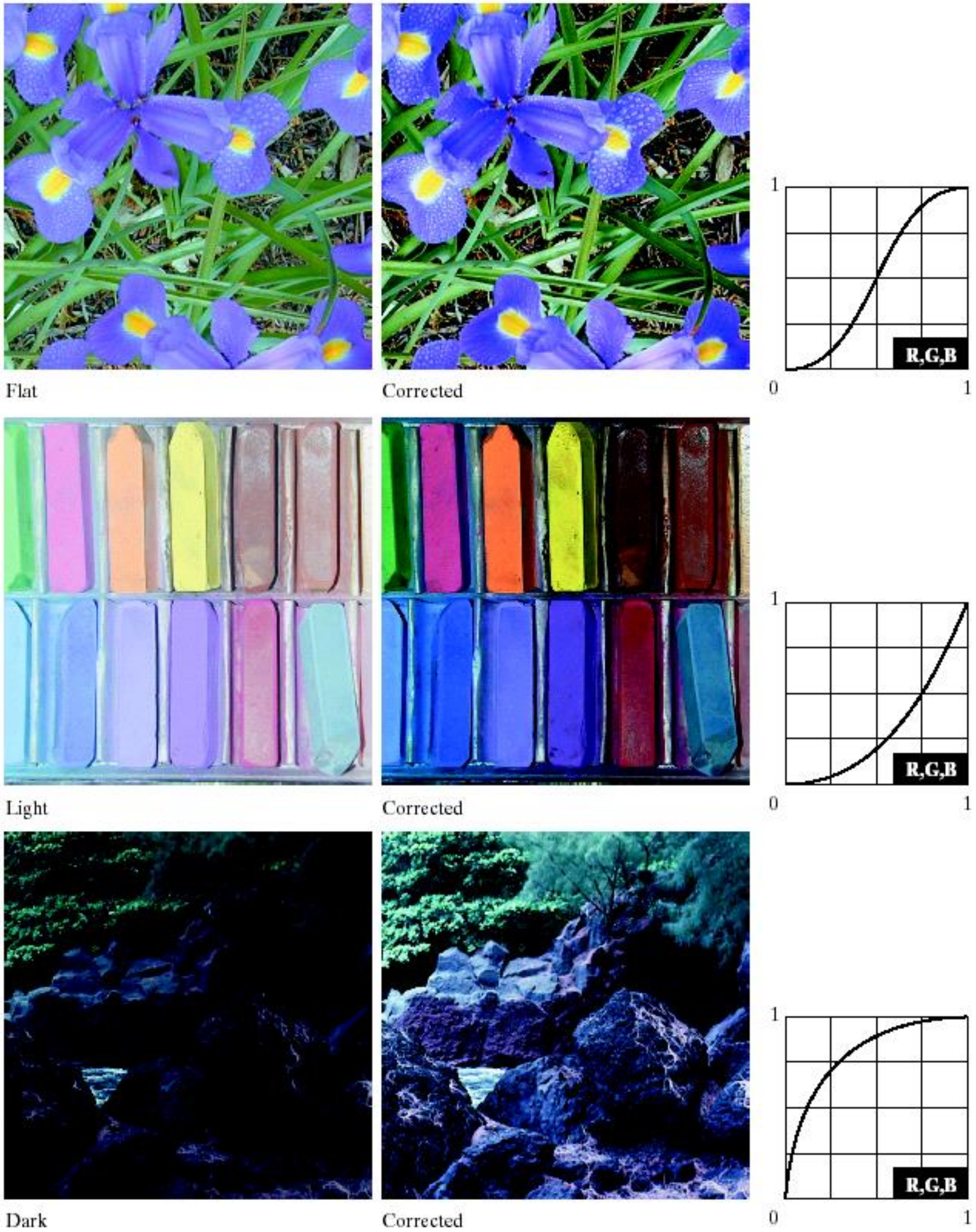
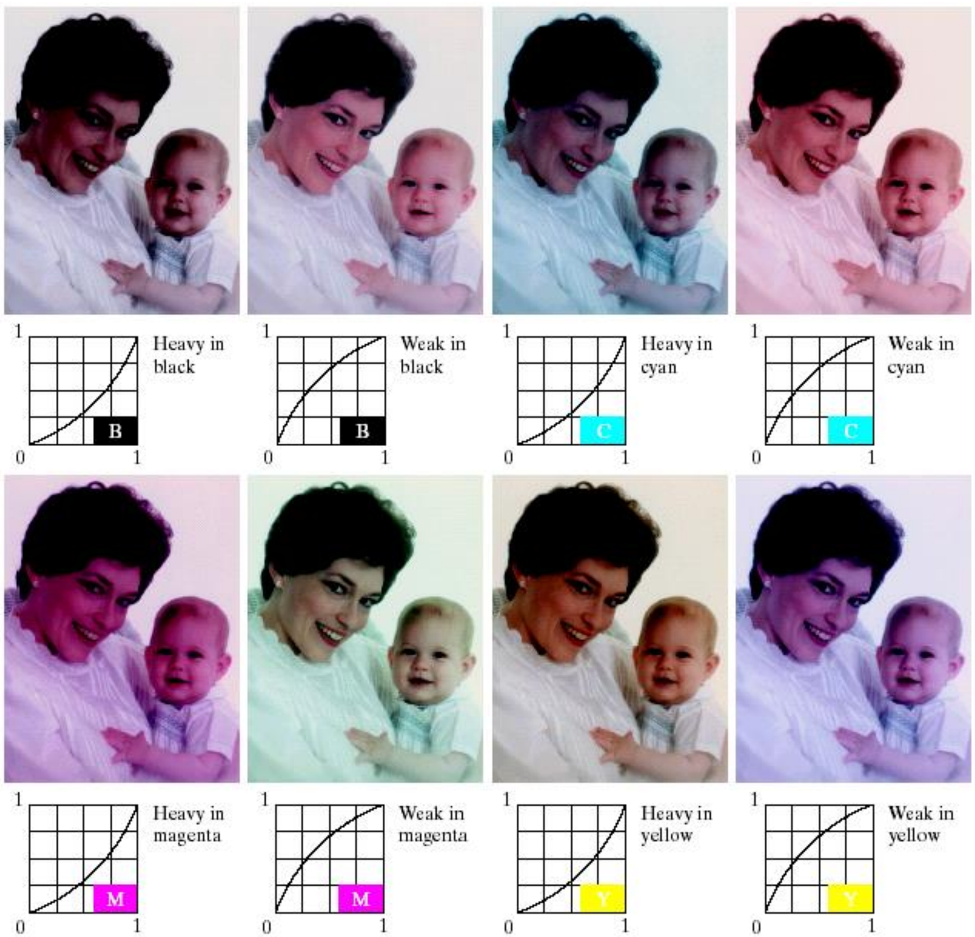


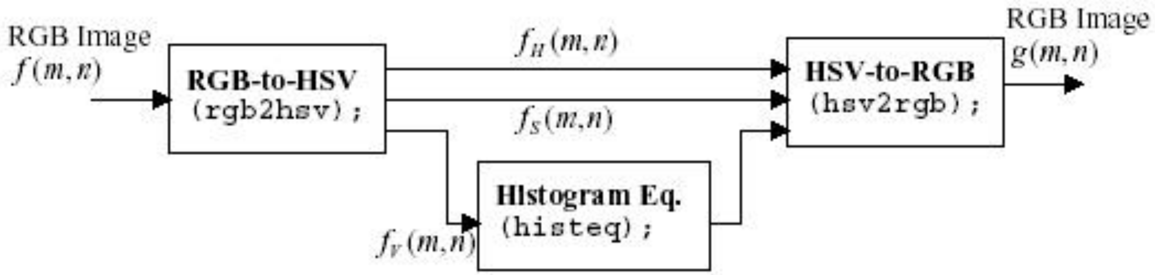
FIGURE 6.33 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Original/Corrected

FIGURE 6.34 Color balancing corrections for CMYK color images.

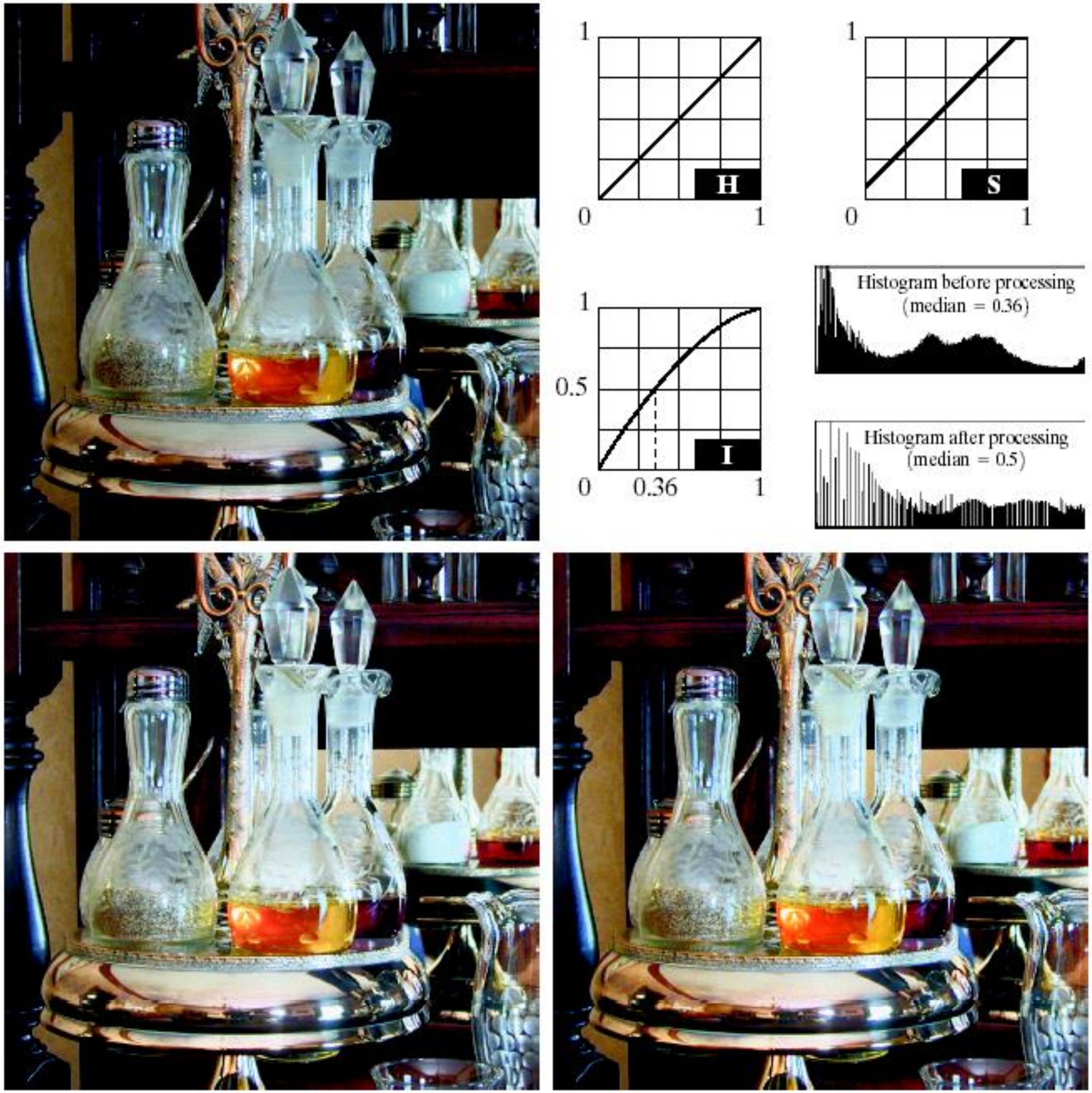


Example: Color Histogram Equalization



Original Image

Histogram Equalized



a b
c d

FIGURE 6.35 Histogram equalization (followed by saturation adjustment) in the HSI colorspace.

Smoothing of Color Images

- Just like the case of grayscale images, smoothing of color images can be performed to remove abrupt transitions of gray values.
- This can be done either in the RGB domain or the HSI domain.
- In the **RGB domain** all three color components are individually transformed by an appropriate smoothing mask, say a 3x3 mask:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

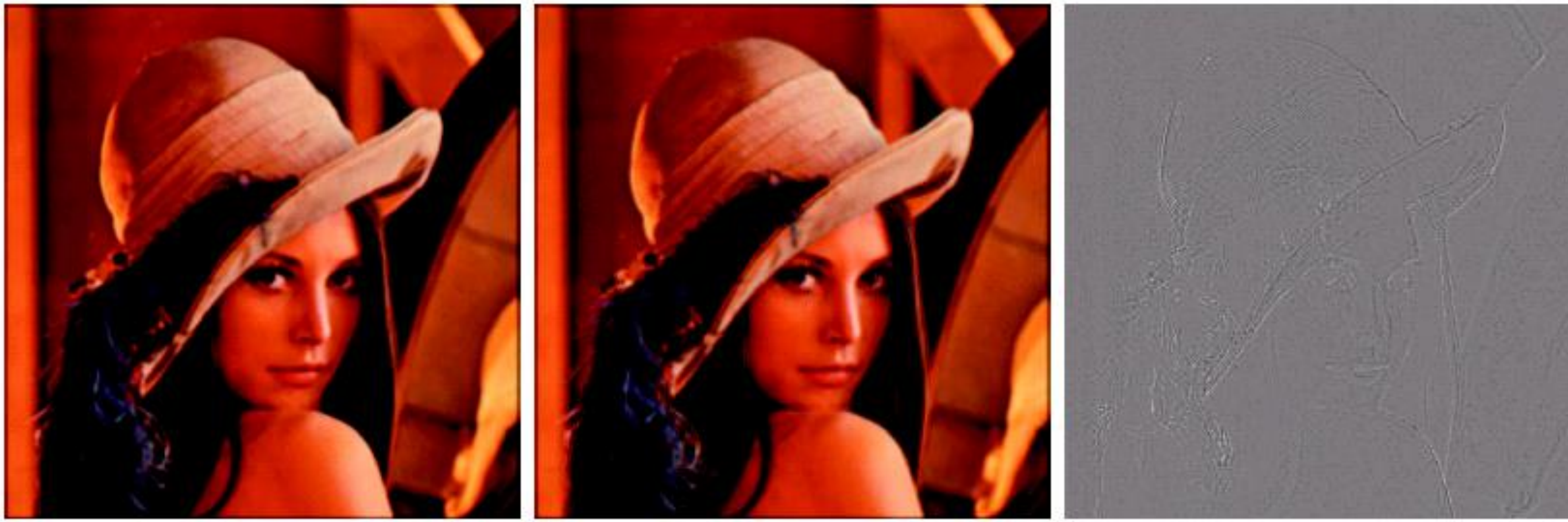
- In the **HSI domain**, **only the I component is transformed** by means of a spatial smoothing mask, leaving the H and S components unchanged.
- In general, the final result in the two cases would **be different**.
  Because the average of two colors is a color intermediate between the two, the former approach has the potential of introducing colors not present in the original image. The latter approach does not have this problem, since the Hue and Saturation components are preserved.

a	b
c	d

FIGURE 6.36 (a) RGB image. (b) Red component image. (c) Green component. (d) Blue component.

a b c

FIGURE 6.37
HSI components of the RGB color image in Fig.6.38(a). (a) Hue. (b) Saturation. (c) Intensity.



a b c

FIGURE 6.38 Image smoothing with a 5 x 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.

Color Image Sharpening

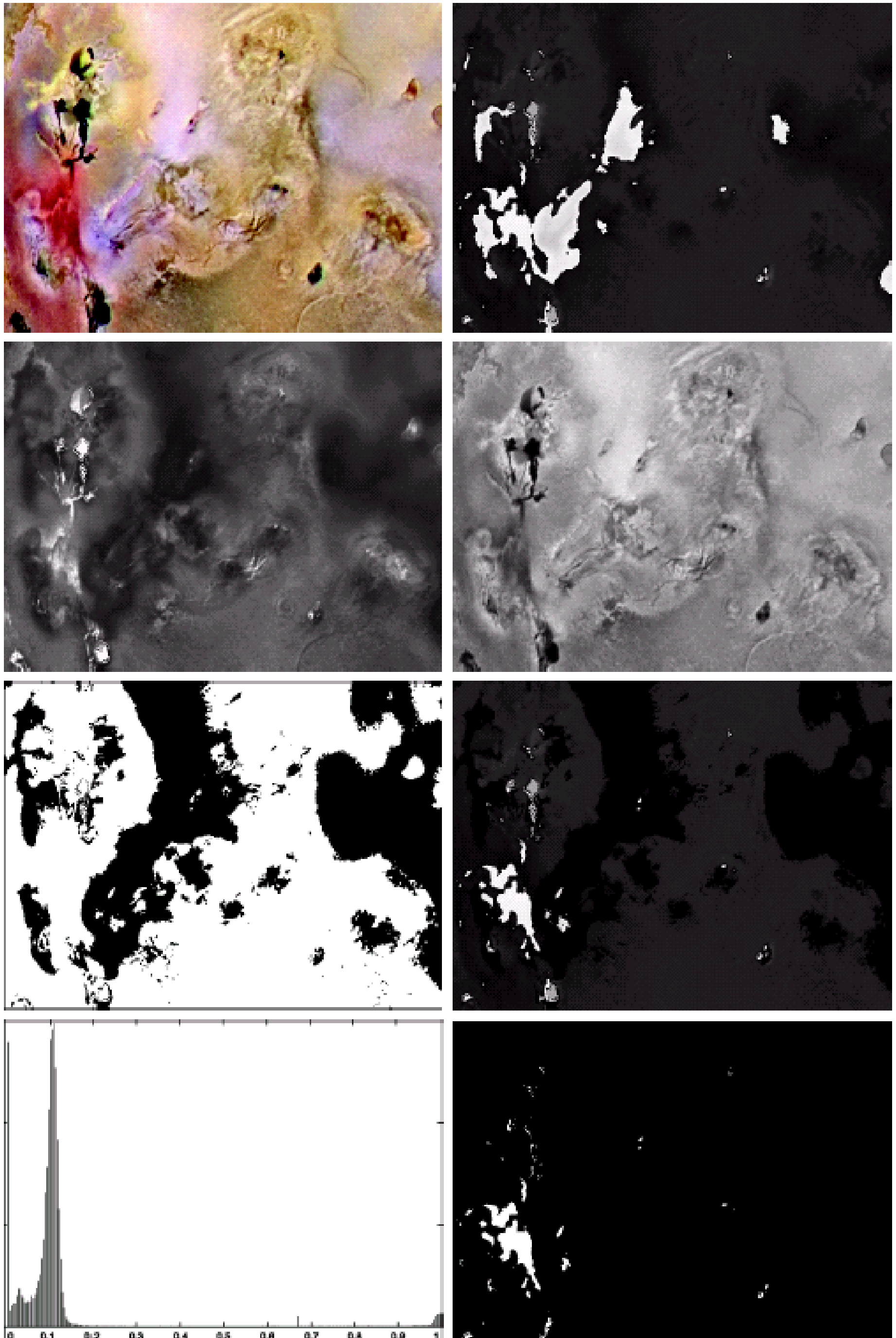
- Sharpening of color images can be performed in a manner analogous to smoothing, using appropriate masks, say the Laplacian mask

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{or} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Example

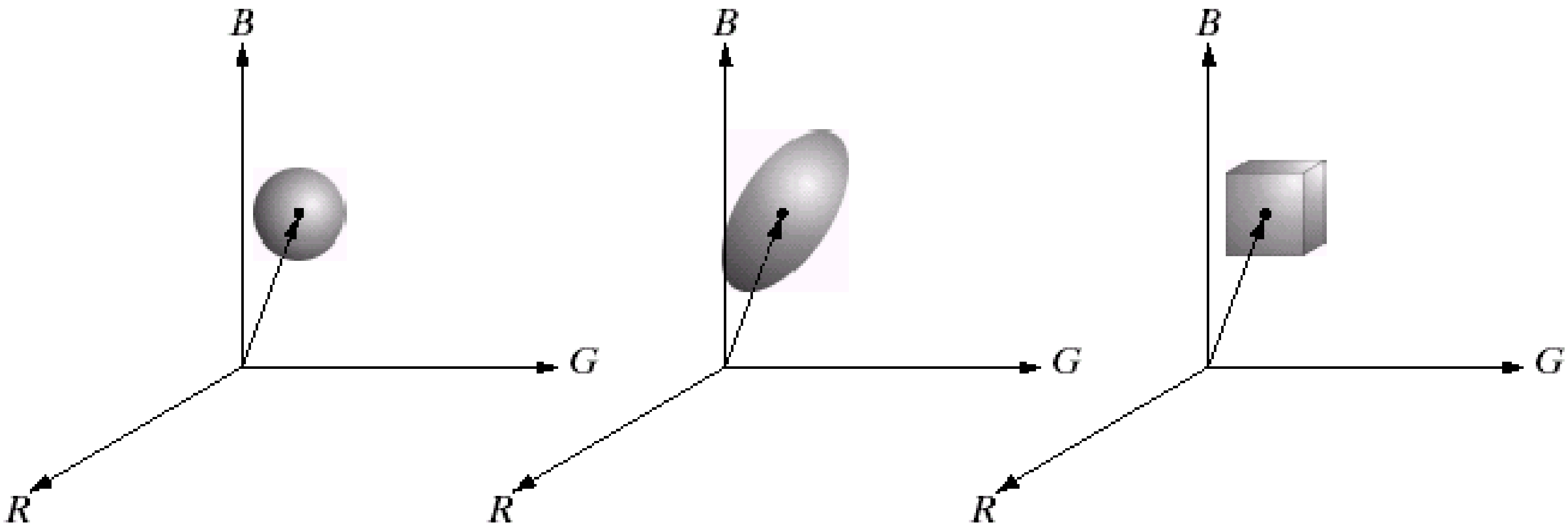
a b c

FIGURE 6.39 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.



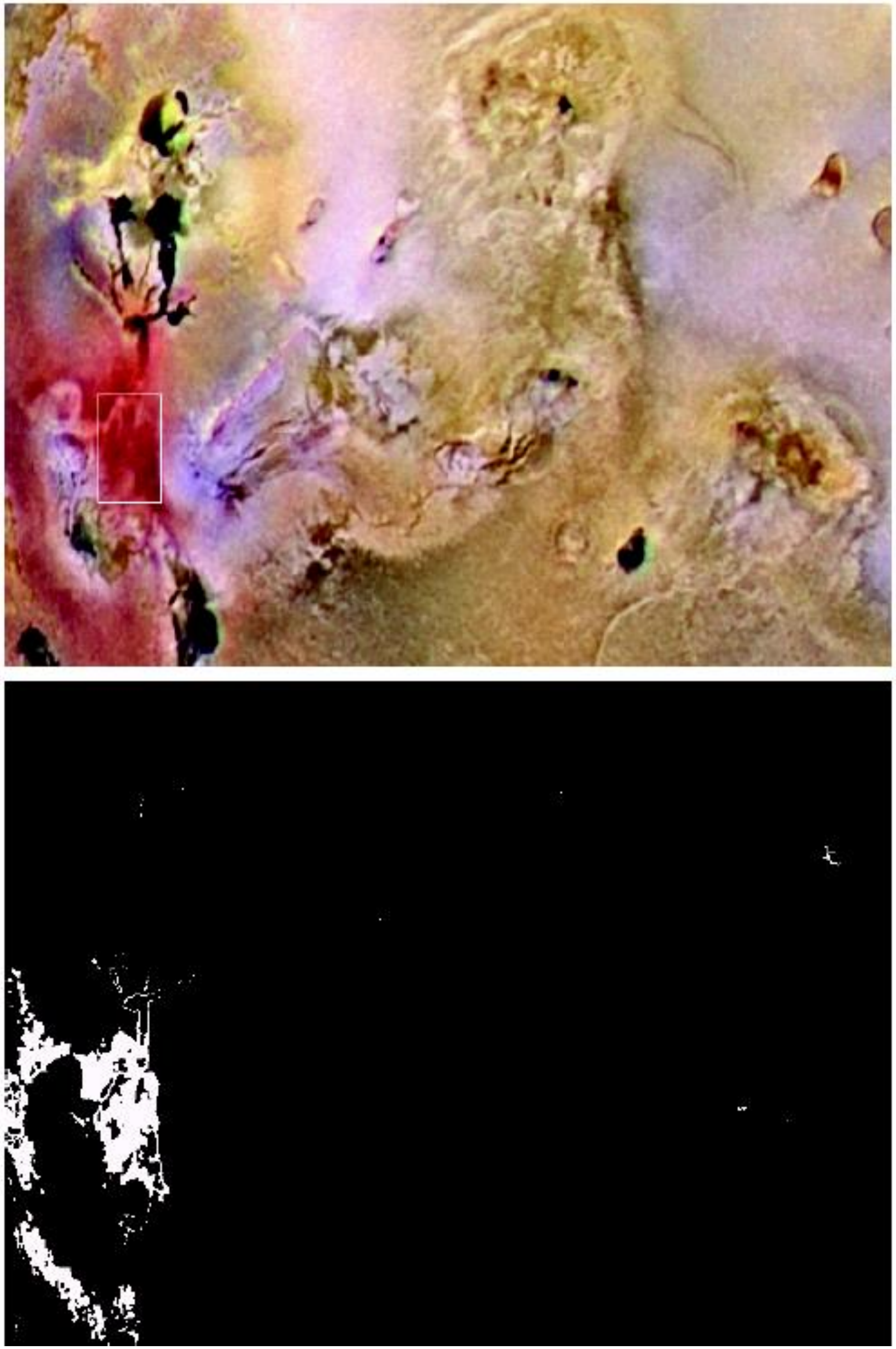
a b
c d
e f
g h

FIGURE 6.40 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 in (a).



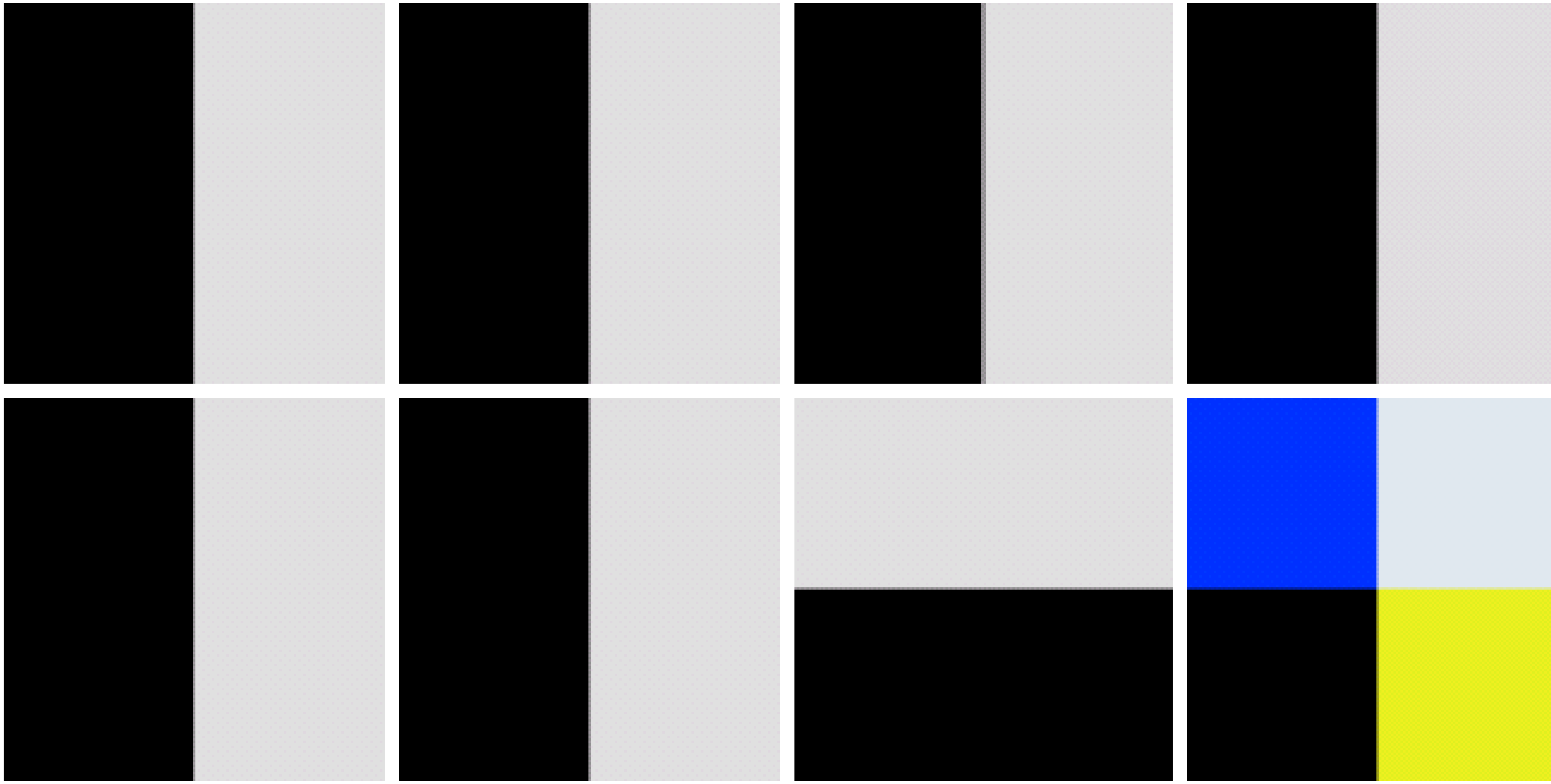
a b c

FIGURE 6.41 Three approaches for enclosing data regions for RGB vector segmentation.



a
b

FIGURE 6.42
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. 6.42(h).

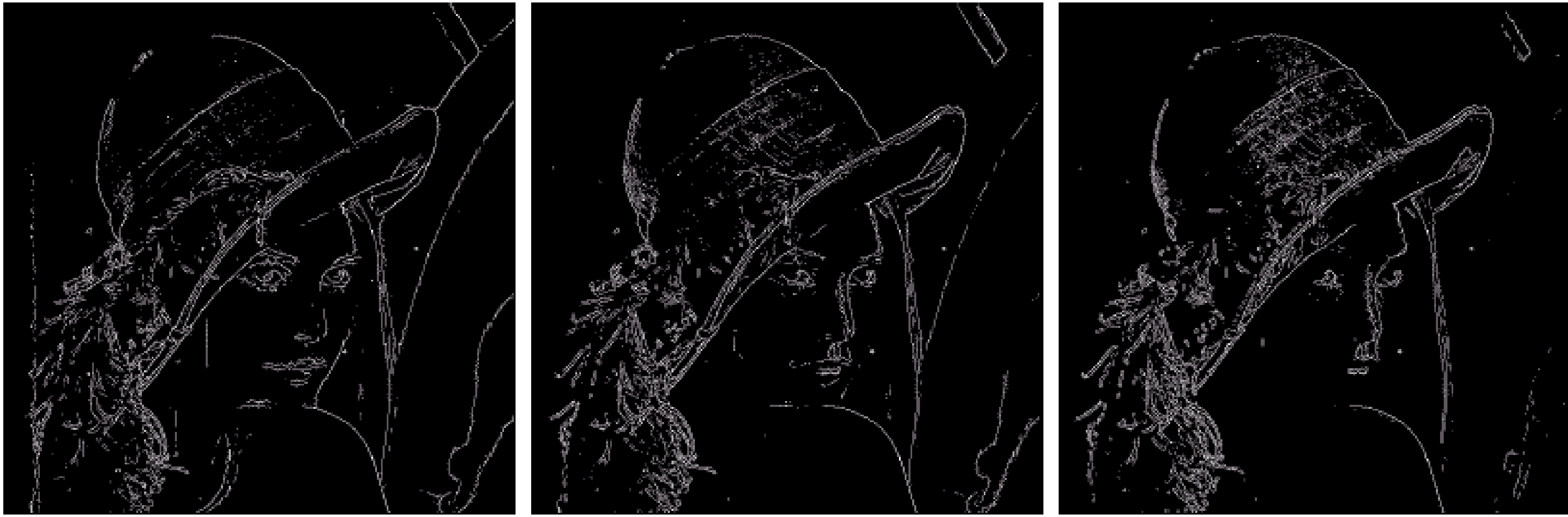


a b c d
e f g h

FIGURE 6.43 (a)-(c) R, G, and B component images and (d) resulting RGB color image. (f)-(g) R, G, and B component images and (h) resulting RGB color image.

a b
c d

FIGURE 6.44
(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).



a b c

FIGURE 6.45 Component gradient images of the color image in Fig.6.44. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 6.44(c).

Noise in color images

- The previously discussed noise models are applicable to color images as well.
- Typically, noise affects all the three color components.
- Usually, across the three color channels, the noise is independent and its statistical characteristic are identical.
- However, due to different illumination conditions or selective malfunction of camera hardware in a particular channel, this may not be the case.
- **Noise filtering by means of a simple averaging can be accomplished by performing the operation independently on the R, G, and B channels and combining the results**.
- However, more complicated filters like the median filter are not as straight-forward to formulate in the color domain and will not be pursued here.
- We will also skip the short discussion in text about color image segmentation and compression.

a b
c d

FIGURE 6.46 (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).]

a b c

FIGURE 6.47 HSI components of the noisy color image in Fig.6.46(d). (a) Hue. (b) Saturation. (c) Intensity.

a b
c d

FIGURE 6.48 (a) RGB image with green plane corrupted by salt-and-pepper noise. (b) Hue component of HSI image. (c) Saturation component. (d) Intensity component.