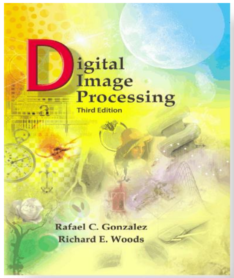
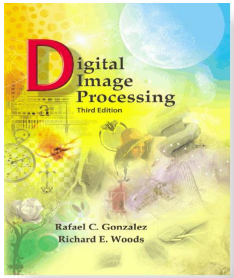
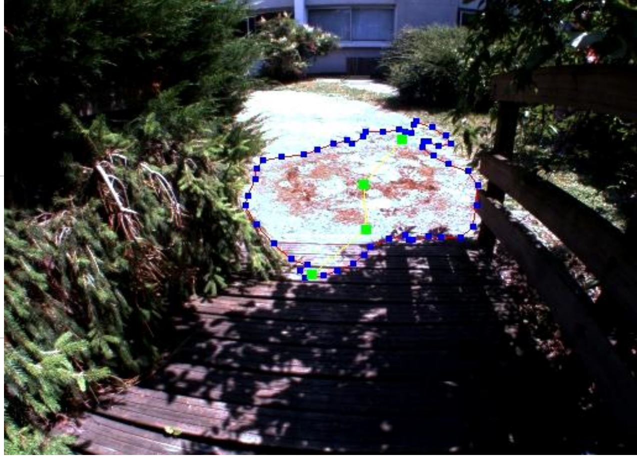


Colour Image Processing

1. Colour Fundamentals
2. Colour Models
3. Pseudocolour Image Processing
4. Basics of Full-Colour Image Processing
5. Colour Transformations
6. Smoothing and Sharpening
7. Image Segmentation based on Colour





Introduction

Motivation to use colour:

- Powerful descriptor that often simplifies object identification and extraction from a scene
- Humans can discern thousands of colour shades and intensities, compared to about only two dozen shades of gray

Two major areas:

- *Full-colour processing*: e.g. images acquired by colour TV camera or colour scanner
- *Pseudo-colour processing*: assigning a colour to a particular monochrome intensity or range of intensities

Some of the gray-scale methods are directly applicable to colour images
Others require reformulation

1. Colour Fundamentals

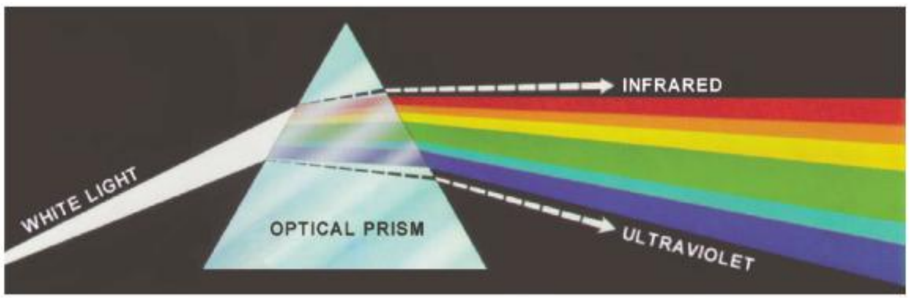


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

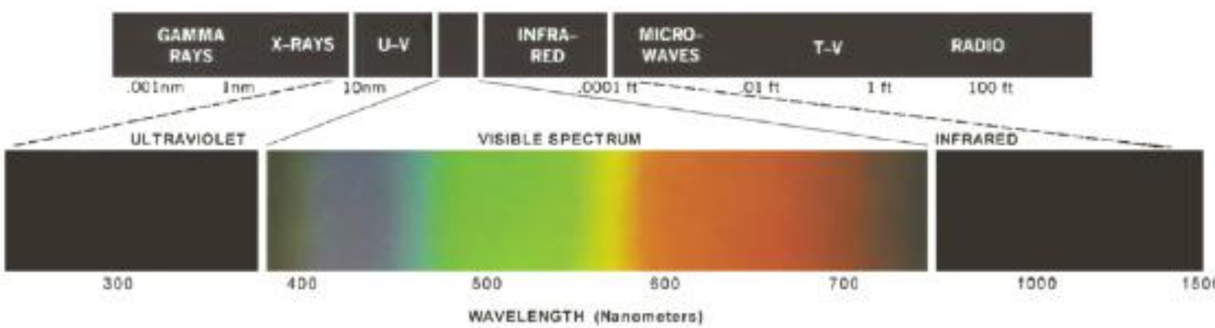


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Perception of colours by the human eye

Cones can be divided into 3 principal sensing categories: (roughly) red, green and blue
~65% are sensitive to red light, ~33% to green light and ~2% to blue (but most sensitive)
⇒ Colours are seen as variable combinations of the *primary* colours: *Red*, *Green*, *Blue*
From CIE* (1931), wavelengths: blue = 435.8nm, green = 546.1nm, red = 700nm

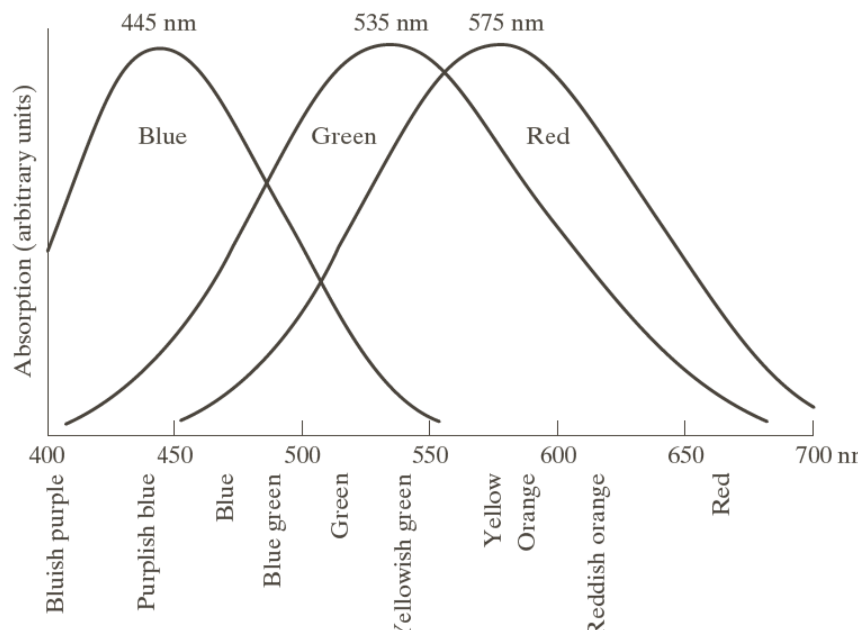
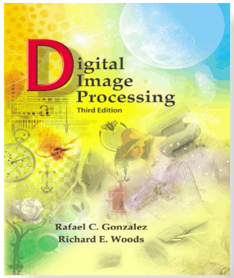


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

* CIE = Commission Internationale de l'Eclairage
(the International Commission on Illumination)

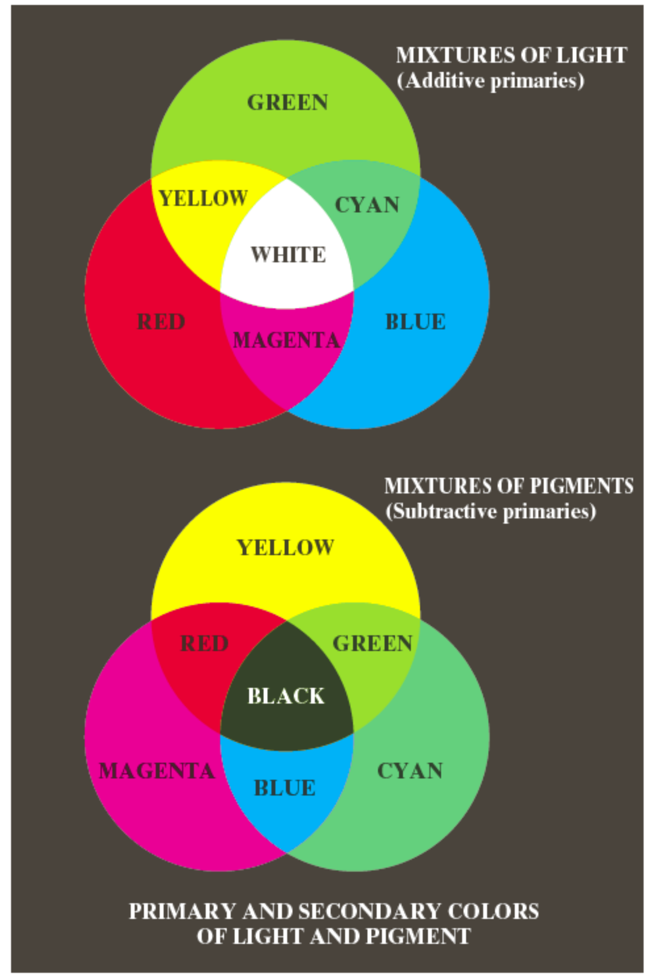


Primary colours can be added to produce the *secondary* colours of light:

- Magenta (red plus blue)
- Cyan (green plus blue)
- Yellow (red plus green)

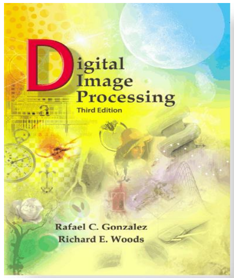
Mixing the three primaries in the right intensities produce white light

Primary colours of pigment: absorb a primary colour of light and reflects or transmits the other two
→ magenta, cyan and yellow



a
b

FIGURE 6.4 Primary and secondary colors of light and pigments. (Courtesy of the General Electric Co., Lamp Business Division.)



Characteristics of a colour:

- **Brightness**: embodies the achromatic notion of intensity
- **Hue**: attribute associated with the dominant wavelength in a mixture of light waves
- **Saturation**: refers to the relative purity or the amount of white light mixed with a hue

(The pure spectrum colours are fully saturated; e.g. Pink (red and white) is less saturated, degree of saturation being inversely proportional to the amount of white light added)

Hue and Saturation together = *chromaticity*

$\Rightarrow$ Colour may be characterized by its brightness and chromaticity

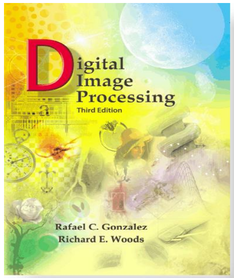
Tristimulus values = amounts of red (X), green (Y) and blue (Z) needed to form a particular colour. A colour can be specified by its trichromatic coefficients:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

NB: $x + y + z = 1$



Content:
62% green
25% red
13% blue

Another approach for specifying colours:
The CIE *chromaticity diagram*:
Shows colour composition as a function of *x* (*red*) and *y* (*green*)

For any value of *x* and *y*: *z* (*blue*) is obtained by:

$$z = 1 - (x + y)$$

Pure colours of the spectrum (fully saturated): boundary

Equal fractions of the 3 primary colours (CIE standard for white light)

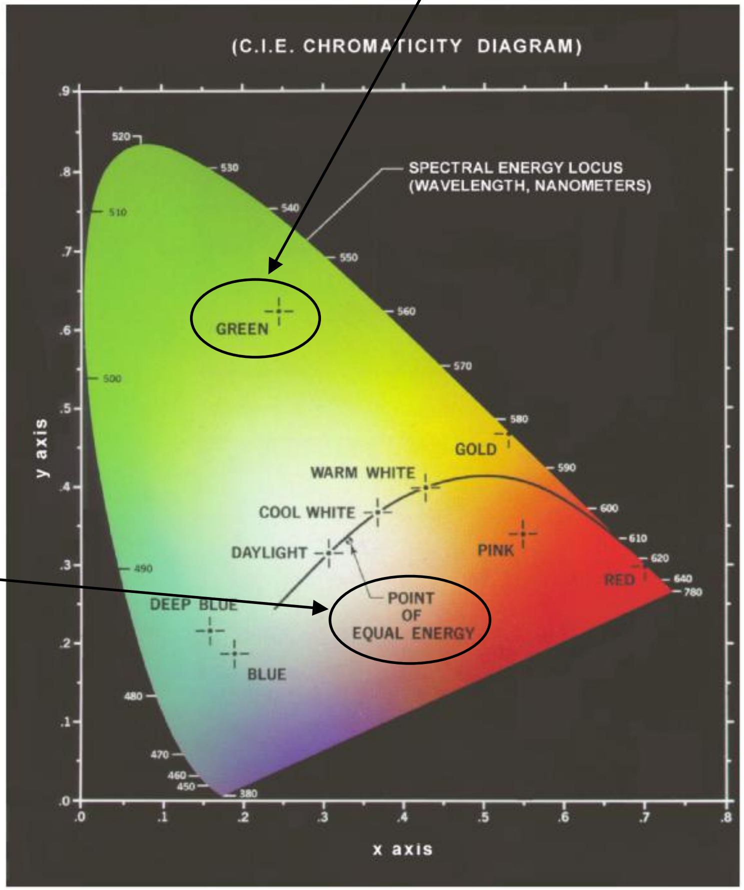


FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)

Typical range of colours (*colour gamut*) produced by RGB monitors:

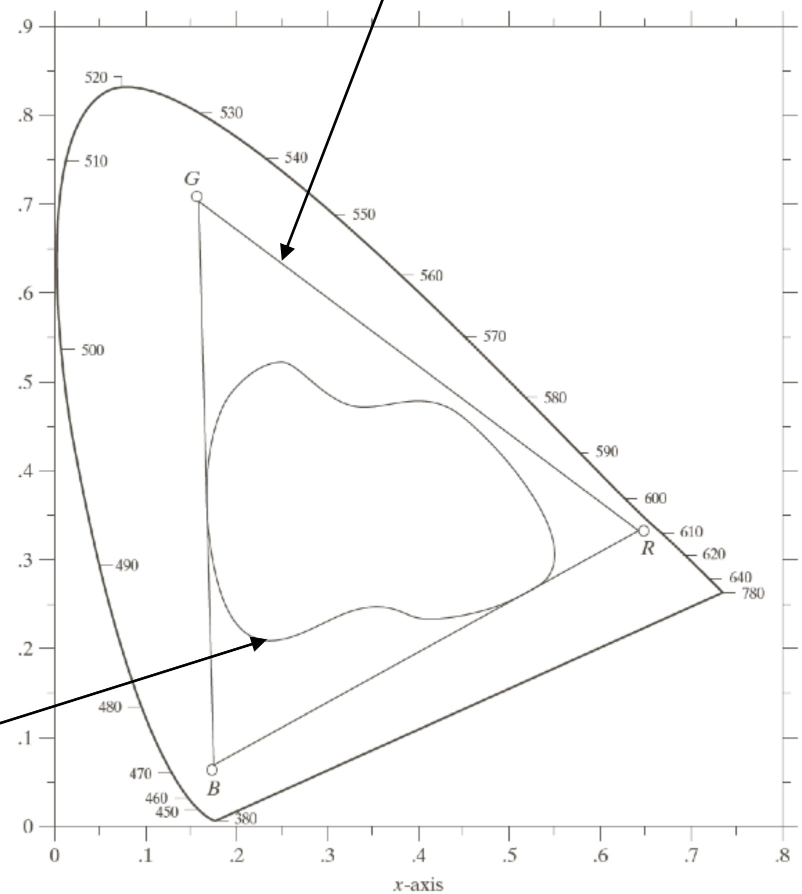
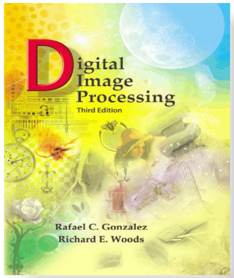


FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

Colour gamut of today's high-quality colour printing devices

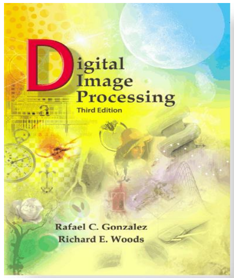


2. Colour Models

- Also called: *colour spaces* or *colour systems*
- Purpose: facilitate the specification of colours in some “standard” way
- Colour model = specification of a coordinate system and a subspace within it where each colour is represented by a single point

Most commonly used hardware-oriented models:

- **RGB** (Red, Green, Blue), for colour monitors and video cameras
- **CMY** (Cyan, Magenta, Yellow) and **CMYK** (CMY+Black) for colour printing
- **HSI** (Hue, Saturation, Intensity)



2.1 The RGB Colour Model

- Each colour appears in its primary spectral components of Red, Green and Blue
- Model based on a Cartesian coordinate System
- Colour subspace = cube
- RGB primary values: at 3 opposite corners (+ secondary values at 3 others)
- Black at the origin, White at the opposite corner

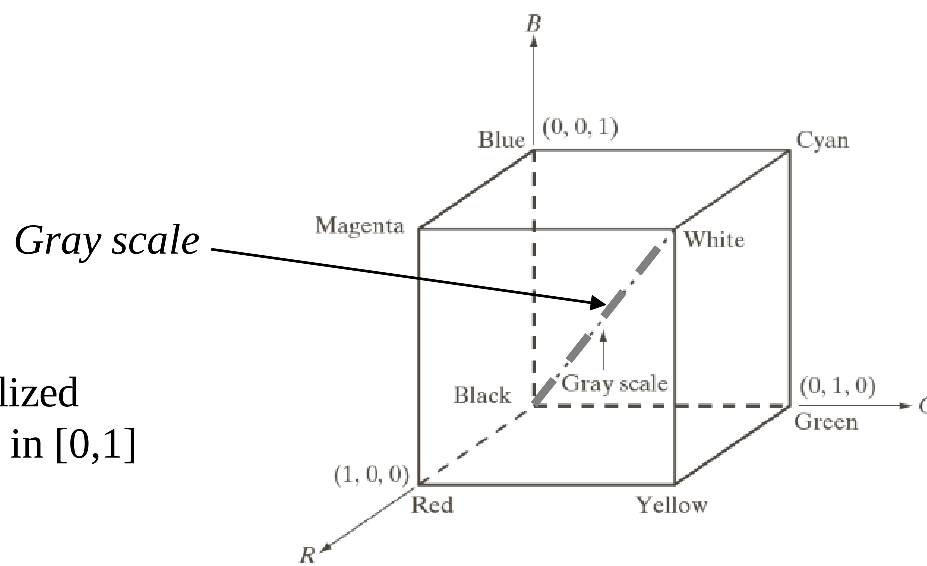


FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

Convention: all colour values normalized
=> unit cube and all values of R,G,B in [0,1]

Number of bits used to represent each pixel in the RGB space = *pixel depth*

Example: RGB image in which each of the red, green and blue images is a 8-bit image

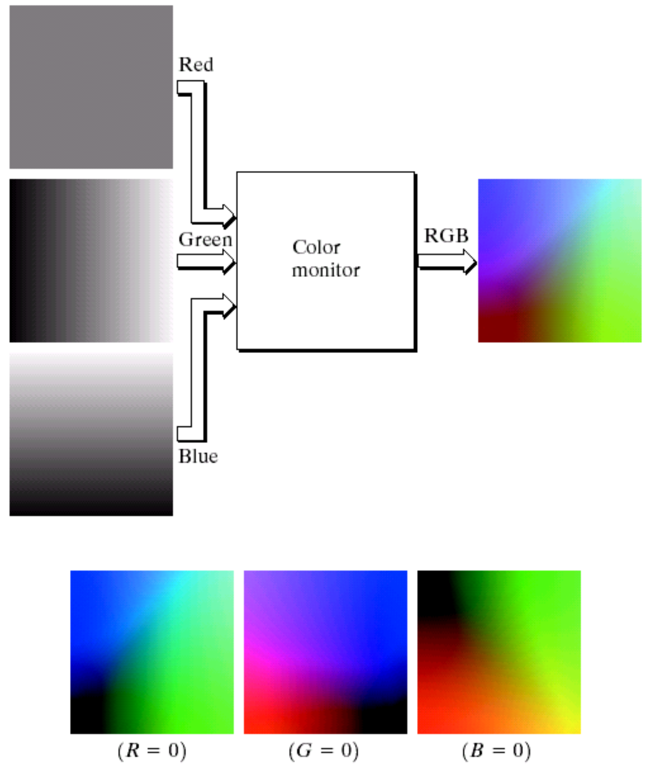
$\Rightarrow$ Each RGB colour pixel (i.e. triplet of values (R,G,B)) is said to have a depth of 24 bits (*full-colour* image)

Total number of colours in a 24-bit RGB image is: $(2^8)^3 = 16{,}777{,}276$

FIGURE 6.8 RGB 24-bit color cube.

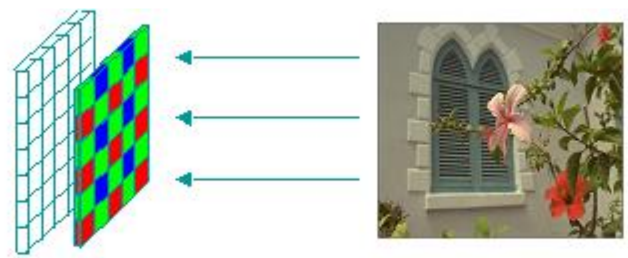
a
b

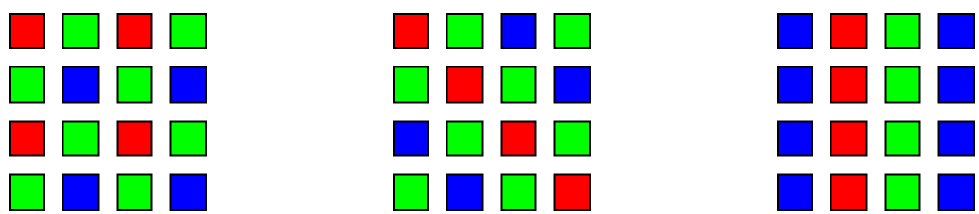
FIGURE 6.9 (a) Generating the RGB image of the cross-sectional color plane (127, G, B). (b) The three hidden surface planes in the color cube of Fig. 6.8.



NB: acquiring an image = reversed process:
Using 3 filters sensitive to red, green and blue, respectively (e.g. Tri-CCD sensor)

Colour Image acquisition: Colour MonoCCD and Bayer Filter Mosaic

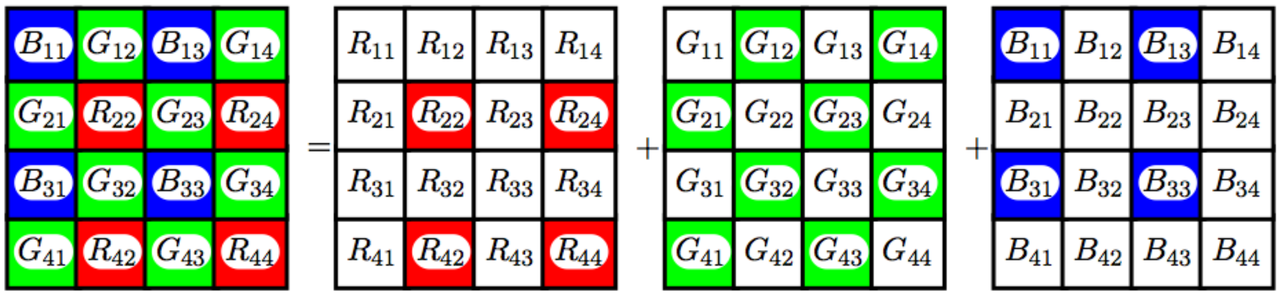




Bayer Mosaic   Diagonal Mosaic   Columns Mosaic

[Aviña Cervantes 2005]

"Demosaicking": Interpolating the values of missing pixels in the component images



Example:

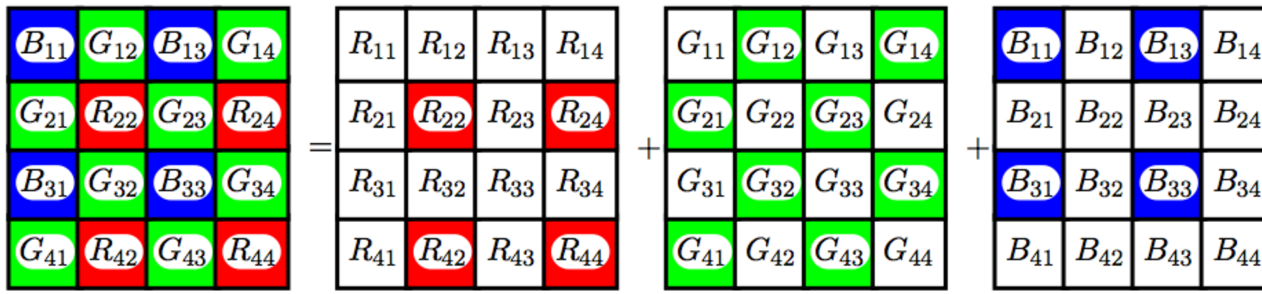
- Bilinear demosaicking:

$$h_G(x) = \left\{ \frac{1}{4}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad x \in \{x_R, x_B\}, \right.$$

$$h_R(x) = \begin{cases} \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} & x \in x_G \\ \frac{1}{4}\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} & x \in x_B, \end{cases} \qquad h_B(x) = \begin{cases} \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} & x \in x_G \\ \frac{1}{4}\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} & x \in x_R \end{cases}$$

[Aviña Cervantes 2005]

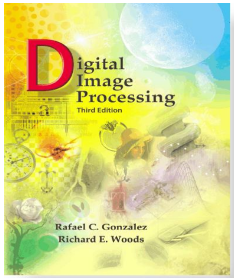
“Demosaicking”



Other methods :

- Nearest Neighbour Method
- Demosaicking by median filter
- Demosaicking by constant hue
- Demosaicking by gradients detection
- Adaptive interpolation by Laplacian

[Aviña Cervantes 2005]

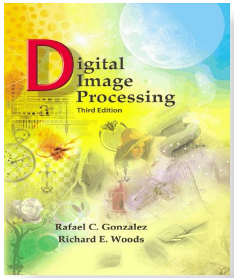


2.2 XYZ (CIE)

- Official definition of the CIE XYZ standard (normalised matrix):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Commonly used form: w/o leading fraction => RGB=(1,1,1) ➔Y=1



2.2 The CMY and CMYK Colour Models

CMY: Cyan, Magenta, Yellow (secondary colours of light, or primary colours of pigments)

• CMY data input needed by most devices that deposit coloured pigments on paper, such as colour printers and copiers

• or RGB to CMY conversion:

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

(assuming normalized colour values)

Equal amounts of cyan, magenta and yellow => black, but muddy-looking in practice
=> To produce true black (predominant colour in printing) a 4^{th} colour, black, is added
=> CMYK model (CMY + Black)

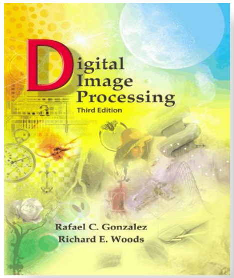
2.3 The HSI Colour Model

RGB and CMY models: straightforward + ideally suited for hardware implementations
+ RGB system matches nicely the human eye perceptive abilities
But, RGB and CMY not well suited for *describing* colours in terms practical for human interpretation

Human view of a colour object described by Hue, Saturation and Brightness (or Intensity)
- *Hue*: describes a pure colour (pure yellow, orange or red)
- *Saturation*: gives a measure of the degree to which a pure colour is diluted by white light
- *Brightness*: subjective descriptor practically impossible to measure. Embodies the achromatic notion of intensity => intensity (gray level), measurable

=> **HSI (Hue, Saturation, Intensity)** colour model

(or HSL: Lightness, HSB: Brightness, HSV: Value)

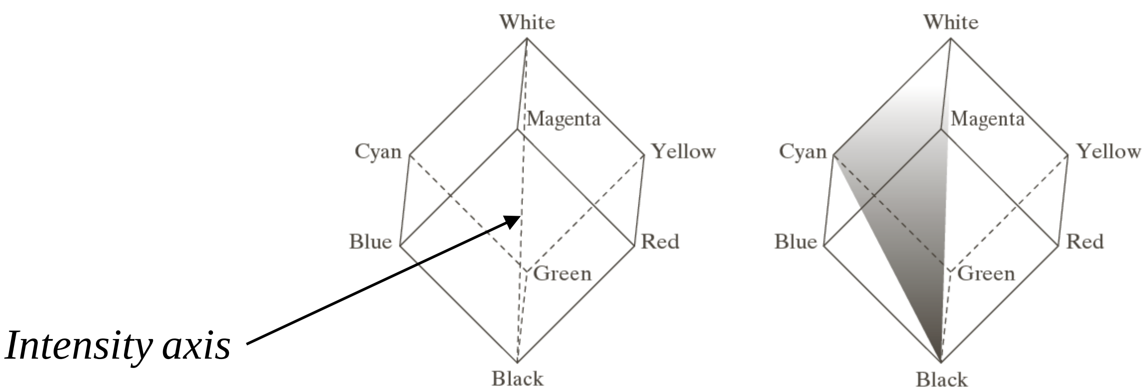


2.3 The HSI Colour Model

Intensity is along the line joining white and black in the RGB cube

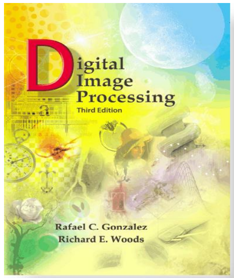
⇒ To determine the intensity component of any colour point: pass a plane perpendicular to the intensity axis and containing the colour point. Intersection of the plane with the axis is the normalized intensity value

⇒ Saturation (purity) of a colour increases as a function of distance from the intensity axis (on the axis, saturation = 0, gray points)

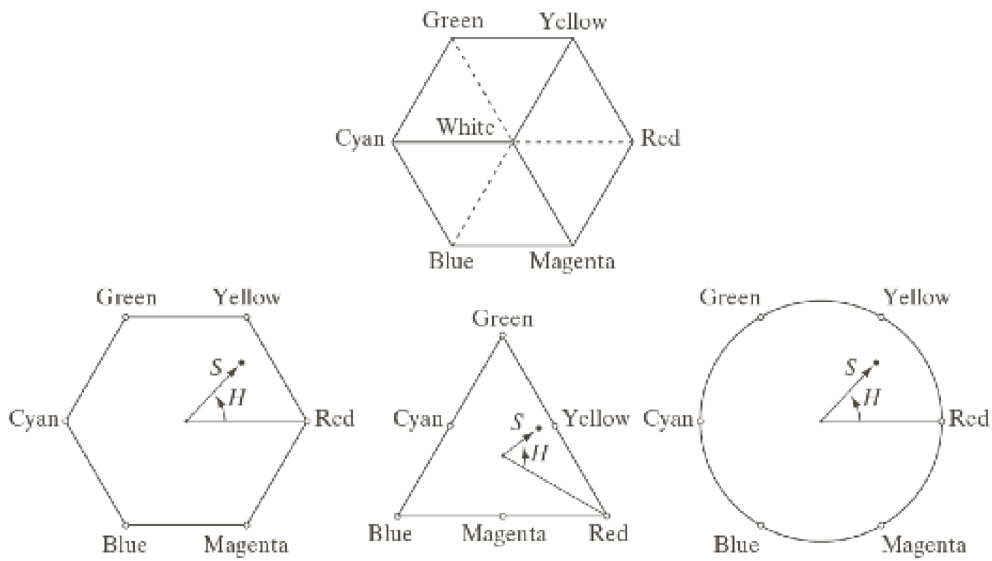


a b

FIGURE 6.12 Conceptual relationships between the RGB and HSI color models.



Colour planes, *perpendicular* to the intensity axis:



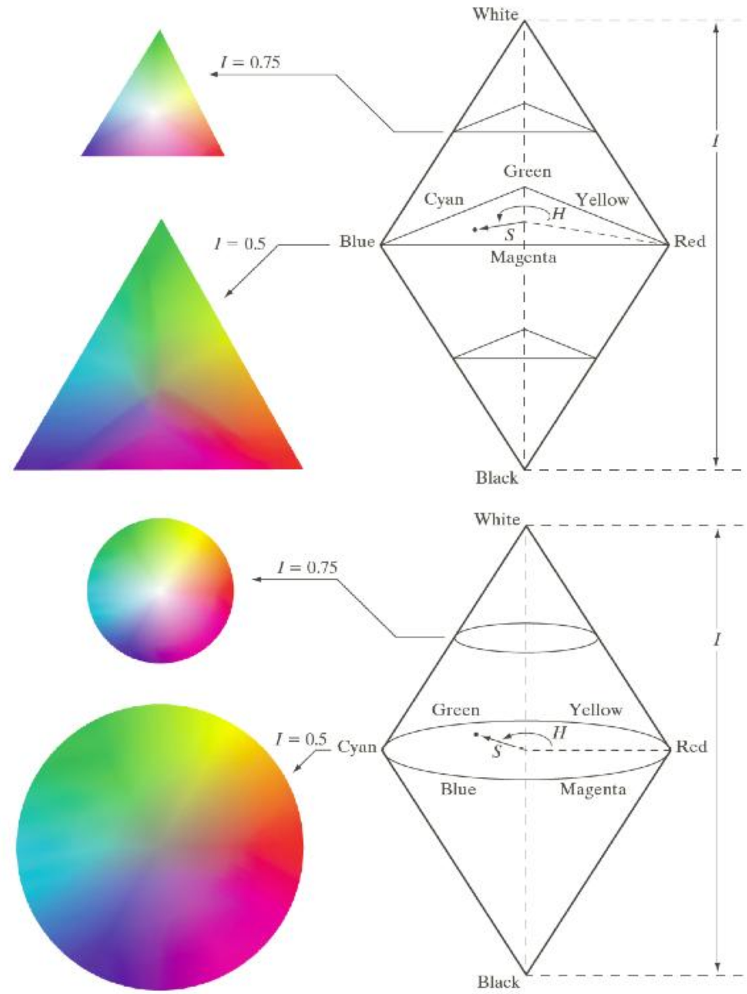
a
b c d

FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

The HSI Colour Models based on:

Triangular colour planes

Circular colour planes



a
b

FIGURE 6.14 The HSI color model based on (a) triangular and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis.

Conversion from RGB to HSI

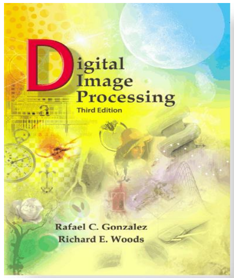
Given an RGB pixel: $$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$ then normalise *H*

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

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

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

NB: RGB values **normalised to [0,1]**,
Theta measured w.r.t. red axis of the HSI space



Conversion from HSI to RGB

Three sectors of interest:

• RG sector ($0° \leq H \leq 120°$):

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

$$R = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

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

• GB sector ($120° \leq H \leq 240°$):

$$H = H - 120°$$

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

$$G = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

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

Conversion from HSI to RGB

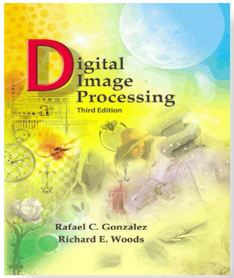
- BR sector (240° ≤ H ≤ 360°): $H = H - 240^\circ$

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

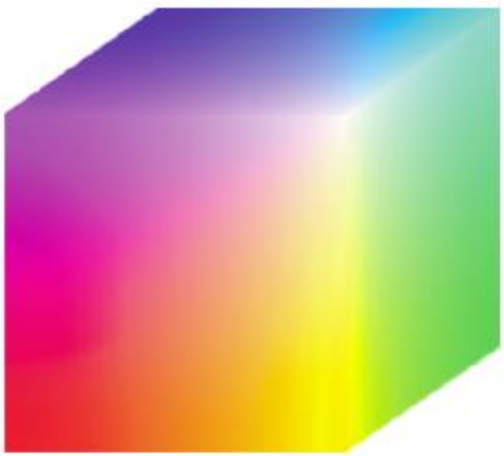
$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

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

- Then normalise *H*



Conversion from HSI to RGB



RGB 24-bit colour cube

=>

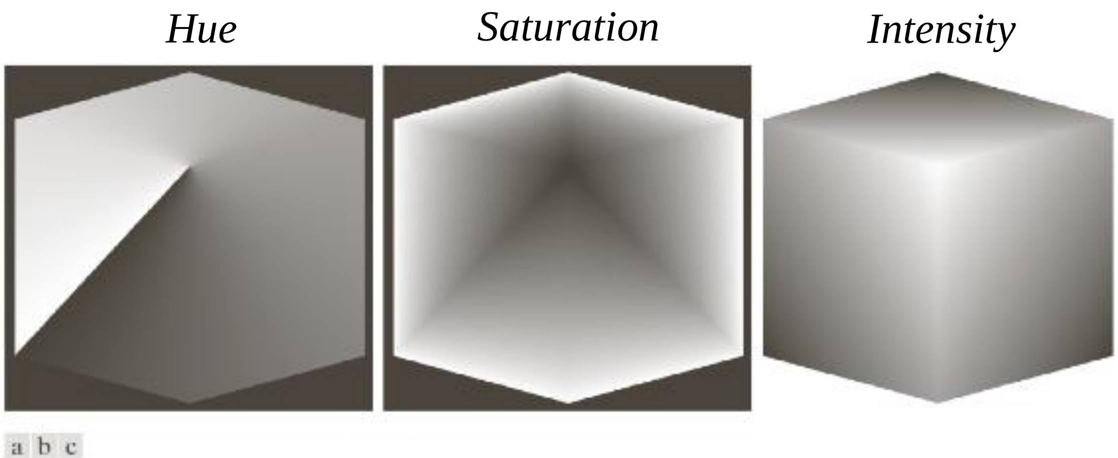
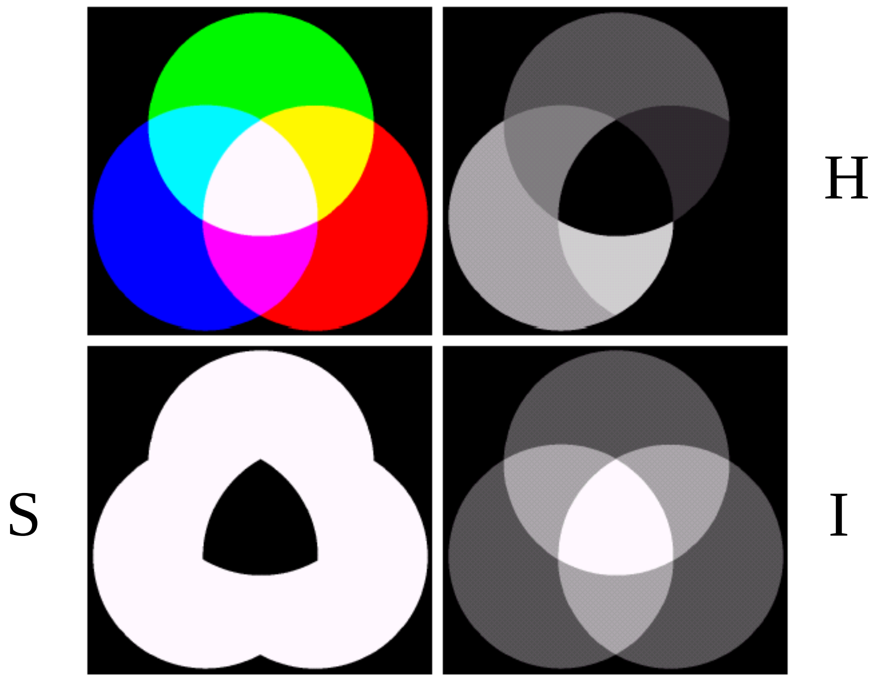


FIGURE 6.15 HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.

Corresponding HSI values

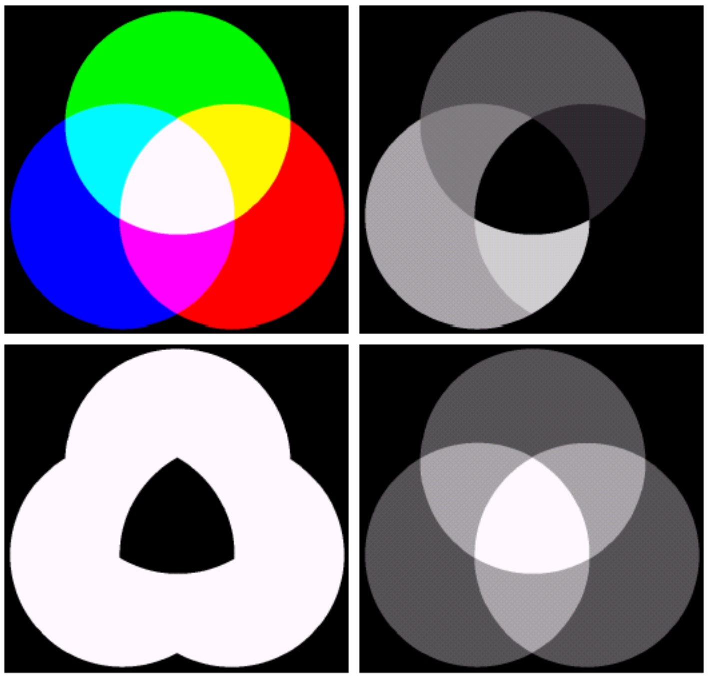
Manipulation of HSI images:

Primary and secondary
RGB colours



a b
c d

FIGURE 6.16 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

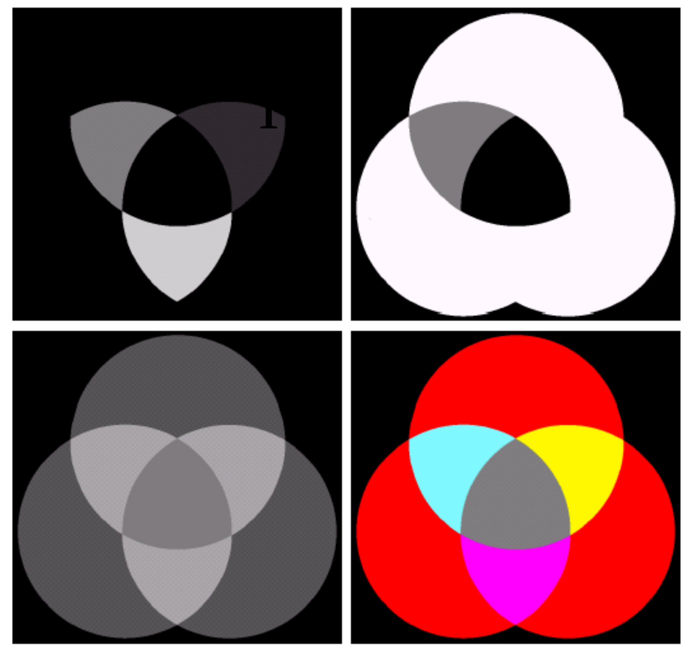


a b
c d

FIGURE 6.16 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

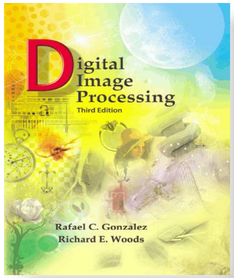
Original image

Modified HSI image



a b
c d

FIGURE 6.17 (a)–(c) Modified HSI component images. (d) Resulting RGB image. (See Fig. 6.16 for the original HSI images.)



2.4 The L*a*b* model

Example of Colour Management System (CMS):
CIE *L*a*b** model, or CIELAB:

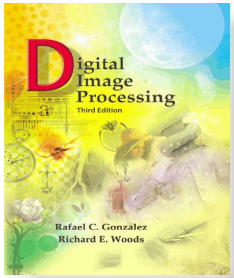
$$L^* = 116\, h\left(\frac{Y}{Y_W}\right) - 16$$

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

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

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

X_W, Y_W and Z_W are *reference white tristimulus*

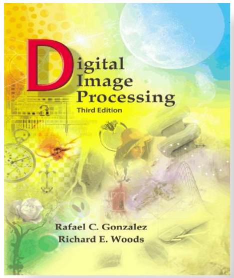


The *L*a*b** colour space is:

- *Colorimetric* (colours perceived as matching are encoded identically)
- *Perceptually uniform* (colour differences among various hues are perceived uniformly)
- *Device independent*

Other characteristics:

- Not a directly displayable format
- Its gamut encompasses the entire visible spectrum
- Can represent accurately the colours of any display, print, or input device
- Like HSI, excellent decoupler of intensity (represented by lightness *L**) and colour (*a** for red minus green, *b** for green minus blue)



4. Basics of Full-Colour Image Processing

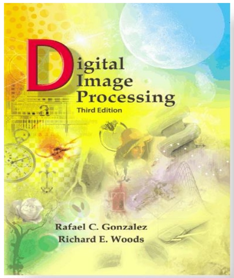
2 major categories:

- Processing of each component image individually
  => composite processed colour image
- Work with colour pixels (vectors) directly

Vector in RGB colour space:

$$\mathbf{c} = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

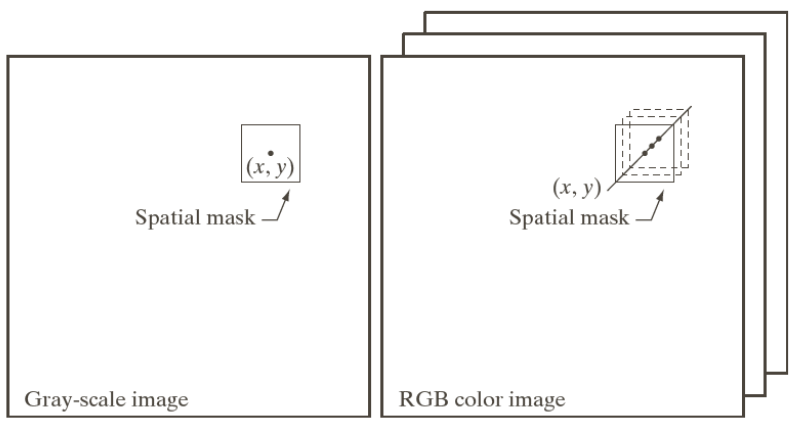
$$\mathbf{c}(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$



4. Basics of Full-Colour Image Processing

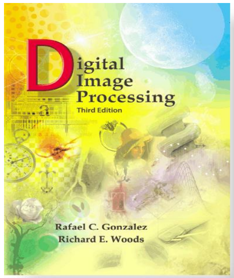
Per-colour-component and *vector-based* processing equivalent iff:

1. The process is applicable to both vectors and scalars
2. The operation on each component of a vector is independent of the other components



a b

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



5. Colour Transformations

NB: Context of a *single* colour model (no conversion between models)

5.1 Formulation

$$g(x,y) = T\left[f(x,y)\right] \qquad \text{(cf. 3.1)}$$

(processed) colour output image → $g(x,y)$

Operator defined over a spatial neighbourhood of point (*x*,*y*) → T

Colour input image → $f(x,y)$

Basic transformations:

$$s_i = T_i(r_1, r_2, \ldots, r_n) \quad i = 1, 2, \ldots, n$$

Set of *transformation* of *colour mapping functions* → T_i

e.g. RGB or HSI: *n=3*. CMYK: *n=4*

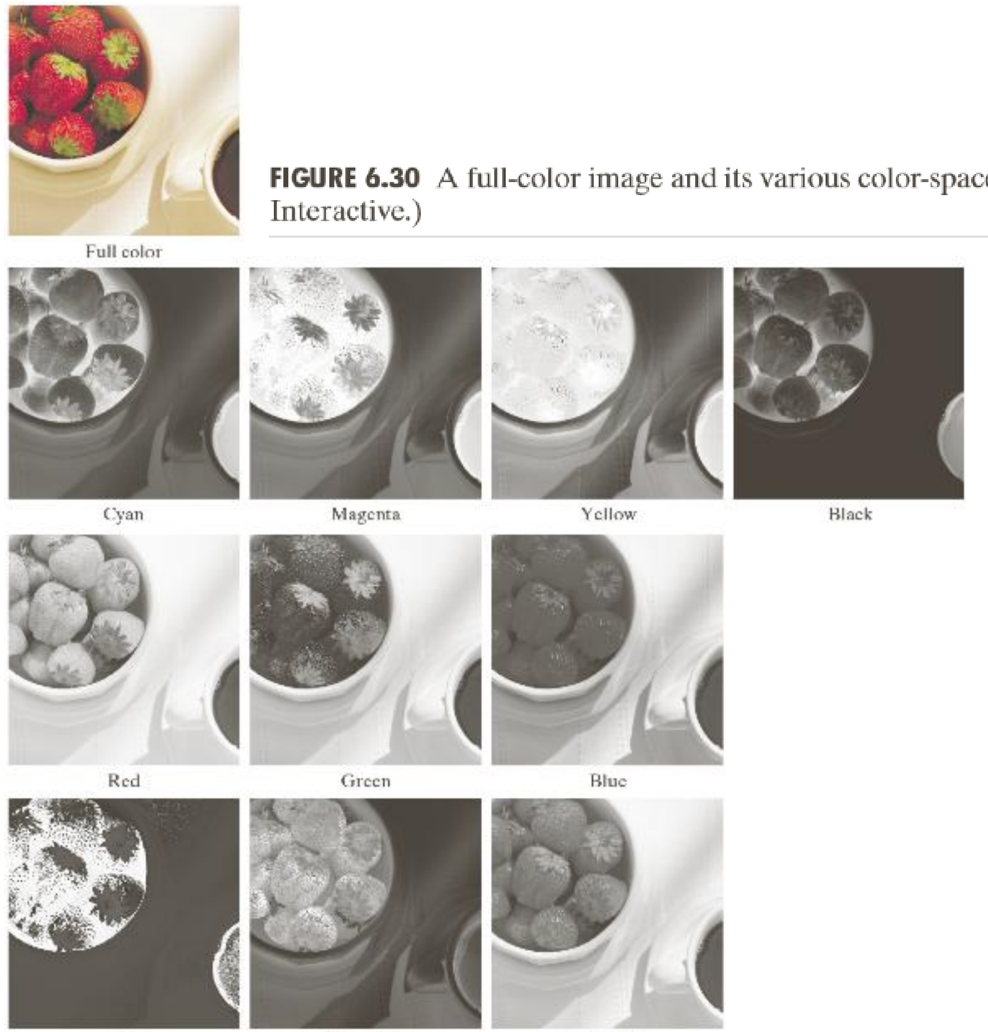


FIGURE 6.30 A full-color image and its various color-space components. Interactive.)

Example: modify the intensity of the full-colour image using: $g(x,y) = k f(x,y)$

$0 < k < 1$

In HSI: $s_3 = k\, r_3 \qquad s_1 = r_1$ and $s_2 = r_2$

In RGB: $s_i = k\, r_i \quad i = 1, 2, 3$

In CMY: $s_i = k\, r_i + (1 - k) \quad i = 1, 2, 3$

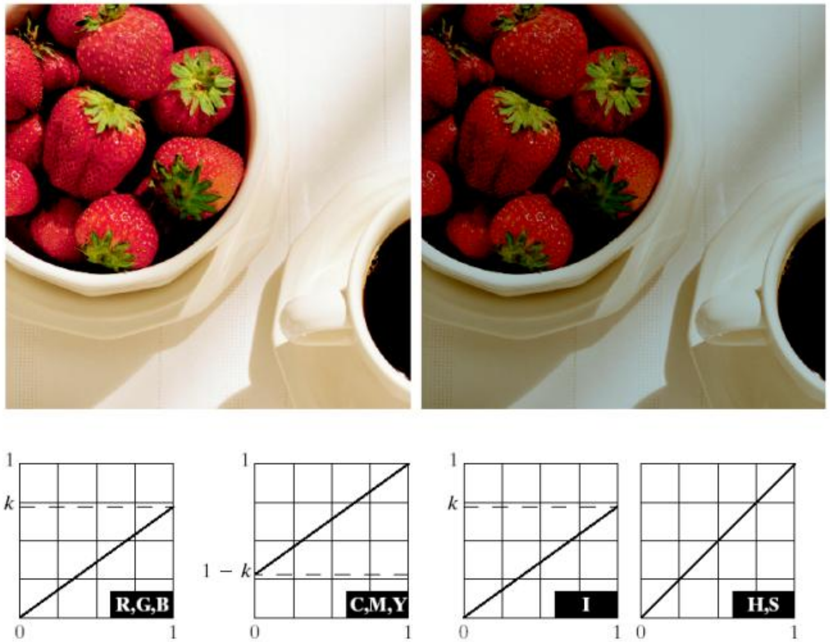


FIGURE 6.31 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)

5.2 Colour complements

Hues directly opposite one another on the *colour circle* = *complements*

Complements are analogous to gray-scale negatives (section 3.2.1)
Useful for enhancing detail embedded in dark regions

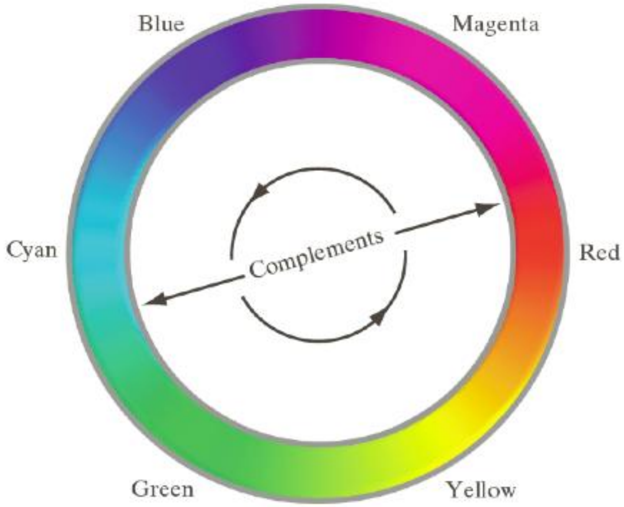
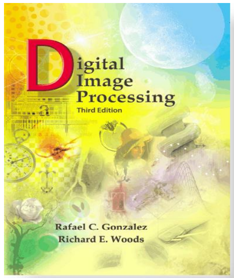
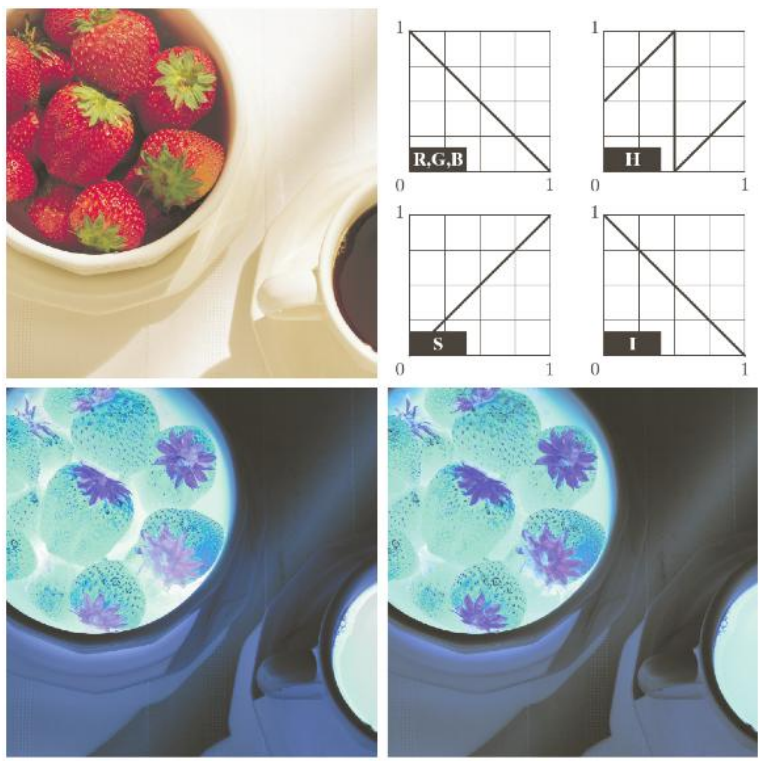


FIGURE 6.32 Complements on the color circle.



5.2 Colour complements



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.

5.3 Colour slicing

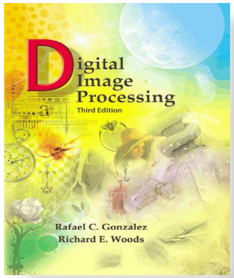
Highlighting a specific range of colours to separate objects from surroundings

- Display the colour of interest, or:
- Use the region defined by colours as a mask

If colours of interest enclosed by a cube (or hypercube) of width W and centered at a prototypical (e.g. average) colours with components $(a_1, a_2, \ldots, a_n)$, then:

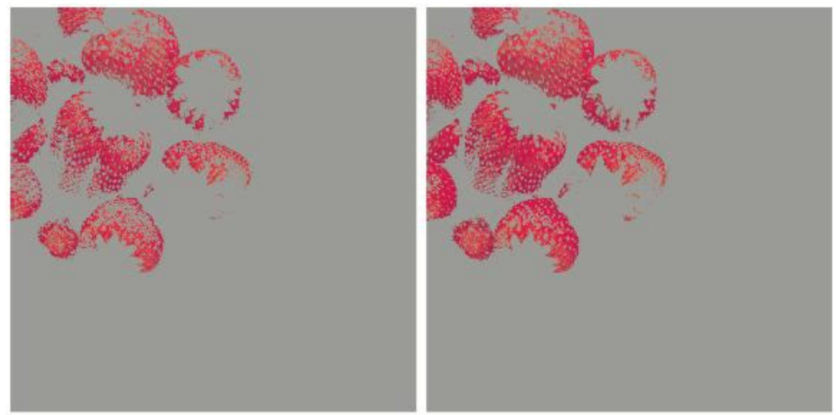
$$s_i = \begin{cases} 0.5 & \text{if } \left[|r_j - a_j| > \frac{W}{2}\right]_{\text{any } 1 \le j \le n} \\ r_i & \text{otherwise} \end{cases} \qquad i = 1, 2, \ldots, n$$

=> Highlight the colours around the prototype by forcing all other colours to the midpoint of the reference colour space (e.g. middle gray in RGB: (0.5,0.5,0.5))



5.3 Colour slicing

=>



a b

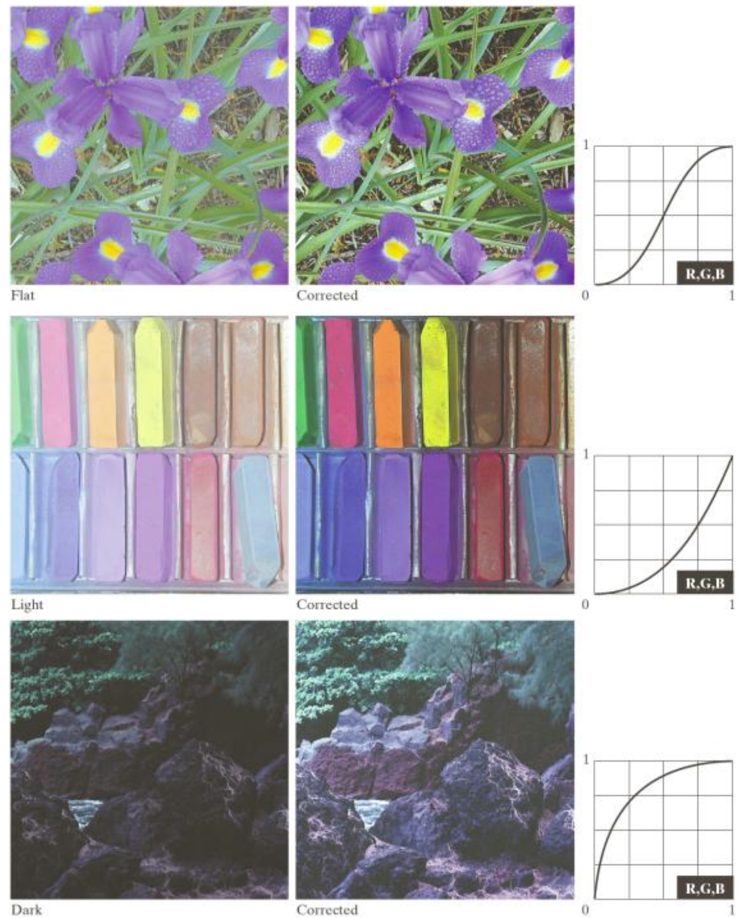
FIGURE 6.34 Color-slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at (0.6863, 0.1608, 0.1922), and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5).

5.4 Tone and Colour Corrections

Effectiveness of these transformations judged ultimately in print
But developed, refined and evaluated on monitors
$\Rightarrow$ Need to maintain a high degree of colour consistency between monitors used and eventual output devices
$\Rightarrow$ *Device-independent* colour model, relating the colour gamuts of the monitors and output devices

1. Tonal transformations

Typical transformations for correcting three common tonal imbalances:



Boosting contrast

Cf. power-law transformations

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.

2. Colour balancing

- Goal: move the white point of a given image closer to pure white (R=G=B)
- Example of strongly coloured illuminant: incandescent indoor lighting (=> yellow or orange hue)
- NB: using white may not always be a good idea…

2. Colour balancing

Analyze (spectrometer) a known colour in an image

When white areas: accurate visual assessments are possible
Other example: skin tones

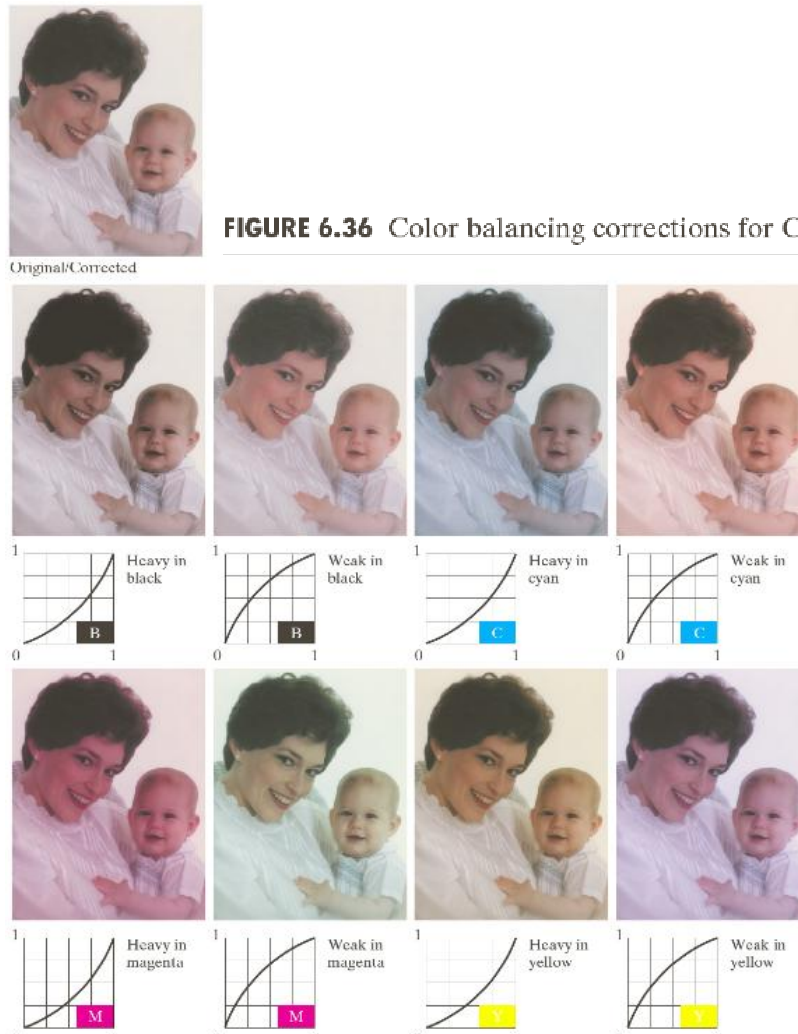
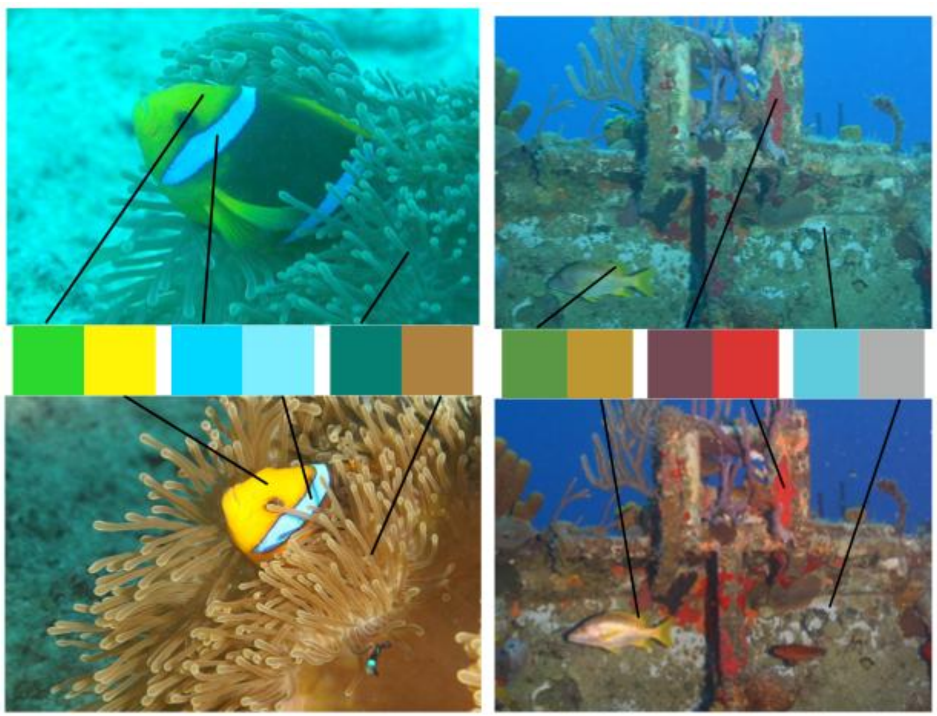


FIGURE 6.36 Color balancing corrections for CMYK color images.

"Colour Adaptation"



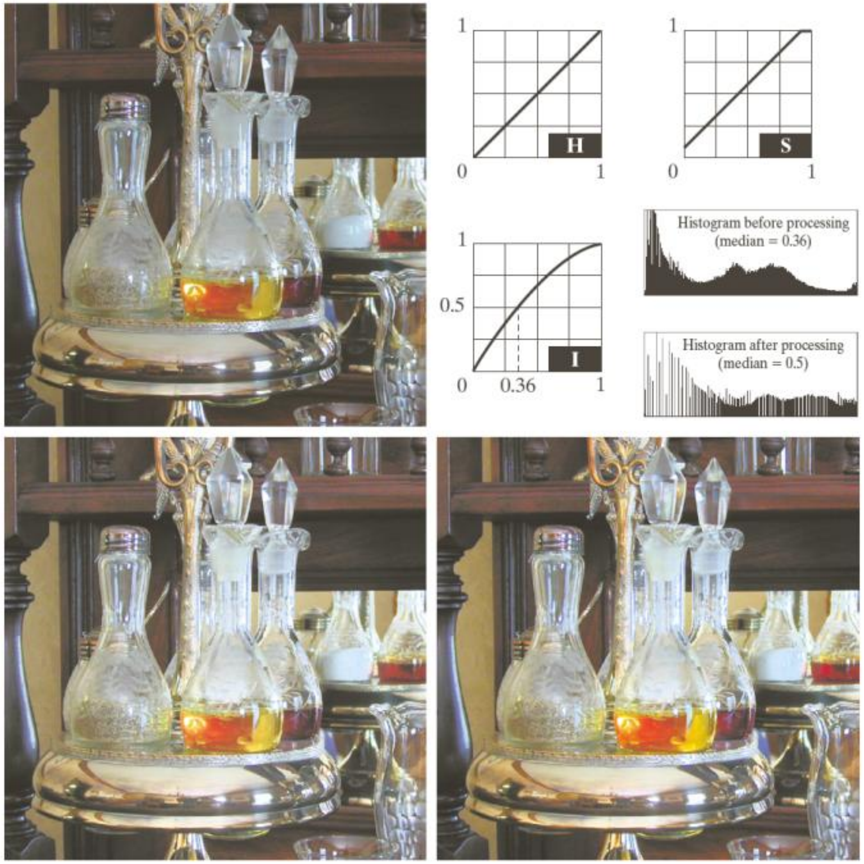
White Flash

"Adaptive Illumination"

Vasilescu et al., *Color-Accurate Underwater Imaging Using Perceptual Adaptive Illumination*, RSS 2010.

5.5 Histogram Processing

Example: Histogram Equalisation in the HSI colour space



a b
c d

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

6 Smoothing and Sharpening

6.1 Colour Image Smoothing

S_{xy} : Set of coordinates of a neighbourhood centered at (x,y) in an RGB image

Average of the RGB component vectors in this neighbourhood:

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

Can be carried out on a per-colour-plane basis (same as averaging using RGB vectors)

Colour Image Smoothing: Example

RGB

R

G

B

a b
c d

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

Colour Image Smoothing: Example

RGB

H S I

a b c

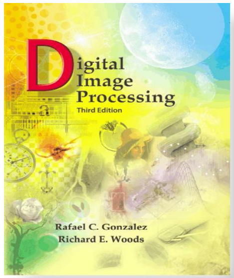
FIGURE 6.39 HSI components of the RGB color image in Fig. 6.38(a). (a) Hue. (b) Saturation. (c) Intensity.

Colour Image Smoothing: Example

Smoothing each RGB component image | *Smoothing the I of HSI* | *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.

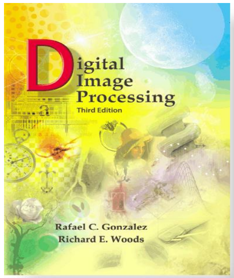


Colour Image Smoothing: Example

Smoothing each RGB component image | *Smoothing the I of HSI* | *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.

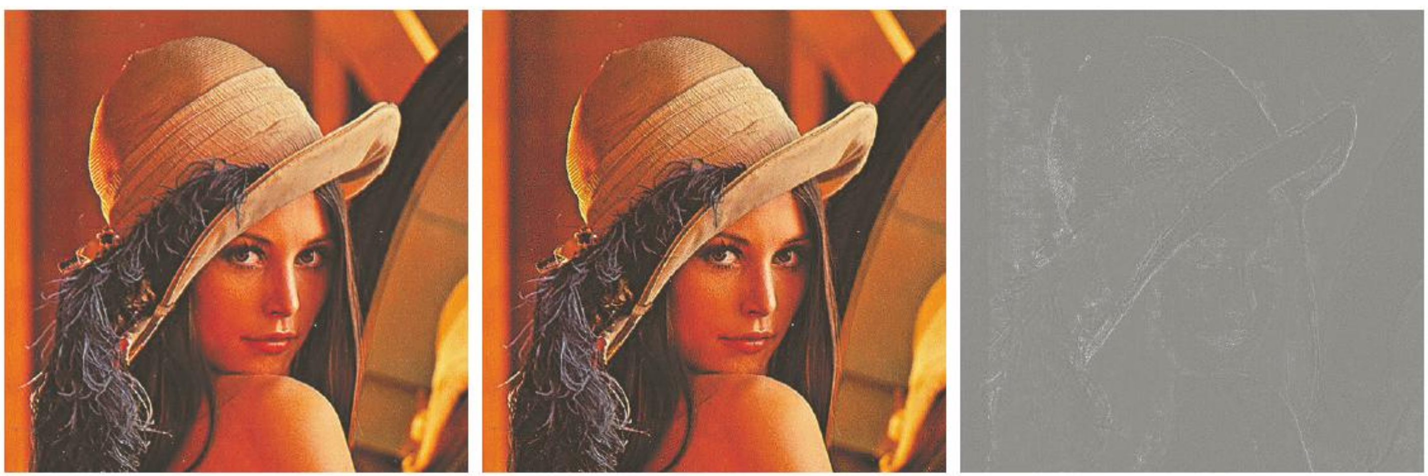


6.1 Colour Image Sharpening

In RGB, the Laplacian of vector **c** is:

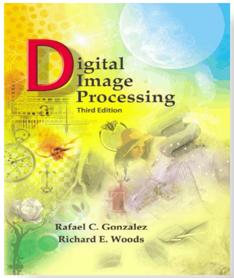
$$\nabla^2 [\mathbf{c}(x, y)] = \begin{bmatrix} \nabla^2 R(x, y) \\ \nabla^2 G(x, y) \\ \nabla^2 B(x, y) \end{bmatrix}$$

=> Can be computed on each component image separately



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.



7 Image Segmentation Based on Colour

7.1 Segmentation in HSI Colour Space

Typically: segmentation on Hue image

Example:

Original image

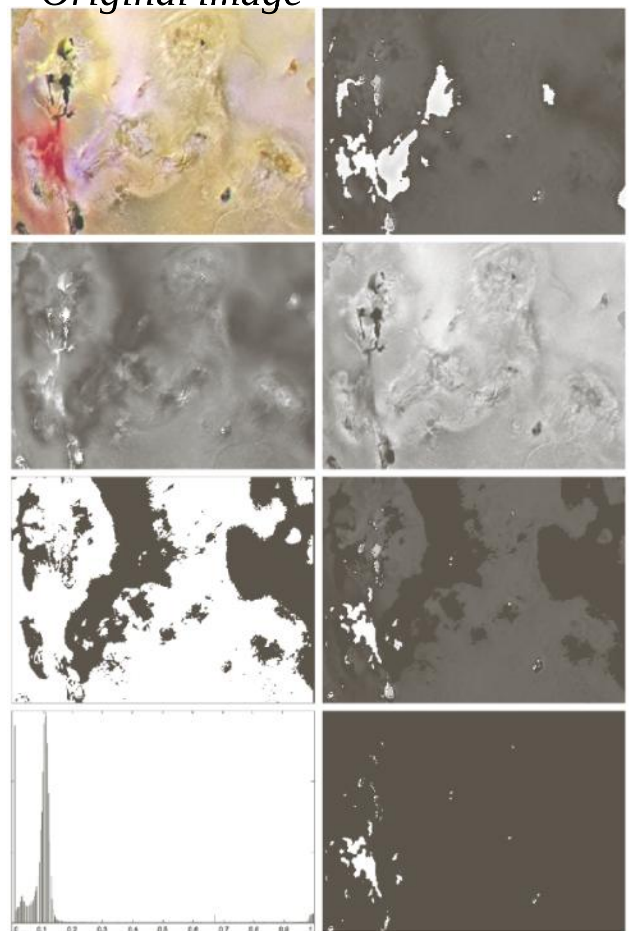
Hue

Saturation

Intensity

Binary saturation mask (threshold=10% of max value)

*Mask * Hue image*



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).

7.2 Segmentation in RGB Colour Space

Given a set of sample colour points of interest, obtain an "average" colour to segment:
=> RGB vector **a**
Segmentation: classify each RGB pixel as having a colour in the specified range or not

z in RGB space is said similar to **a** if the distance between them is less than a specified threshold
Euclidean distance:

$$\begin{aligned} D(\boldsymbol{z},\boldsymbol{a}) &= \|\boldsymbol{z}-\boldsymbol{a}\| \\ &= \left[(\boldsymbol{z}-\boldsymbol{a})^T(\boldsymbol{z}-\boldsymbol{a})\right]^{1/2} \\ &= \left[(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2\right]^{1/2} \end{aligned}$$

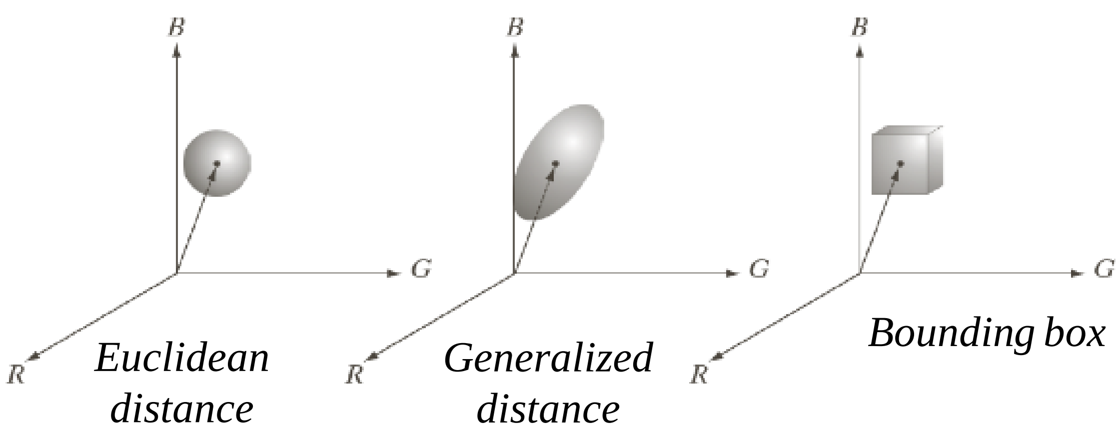
Segmentation criteria: $D(\boldsymbol{z},\boldsymbol{a}) \leq D_0$

7.2 Segmentation in RGB Colour Space

Generalization of the distance measure: $D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T \mathbf{C}^{-1}(\mathbf{z}-\mathbf{a})\right]^{1/2}$

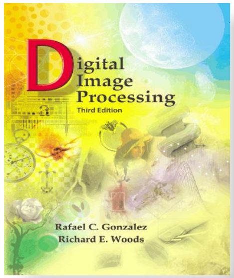
(*Mahalanobis distance*)

Where **C** is the covariance matrix of the samples representative of the colour we wish to segment



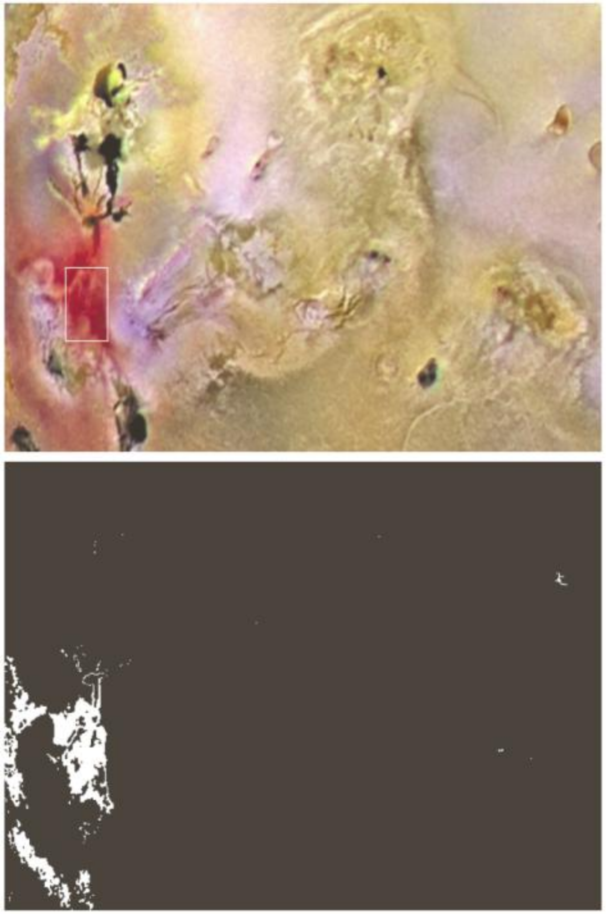
a b c

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



Example:

1. Compute mean vector a of colour points in rectangle
2. Compute the standard deviation of red, green and blue values
3. Box centered on a, dimensions along each RGB axes: 1.25 σ_R



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).

References:

- R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd Edition, Prentice Hall, 2008
- D.A. Forsyth and J. Ponce, *Computer Vision – A Modern Approach*, Prentice Hall, 2003
- J.G. Aviña Cervantes, *Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré* (*Visual Navigation of a Mobile Robot in semi-structured outdoor environment*), PhD Thesis, Institut National Polytechnique de Toulouse, 2005