Colour Image Processing

ICT4201:DIP

Preview

- Why use colour in image processing?
  - colour is a powerful descriptor
    - Object identification and extraction
    - eg. Face detection using skin colours
  - Humans can discern thousands of colour shades and intensities
    - c.f. Human discern only two dozen shades of grays

- Two category of colour image processing
  - Full colour processing
    - Images are acquired from full-colour sensor or equipments
  - Pseudo-colour processing
    - In the past decade, colour sensors and processing hardware are not available
    - colours are assigned to a range of monochrome intensities

Outline

- colour fundamentals
- colour models
- Pseudo-colour image processing
- Basics of full-colour image processing
- colour transformations
- Smoothing and sharpening

colour fundamentals

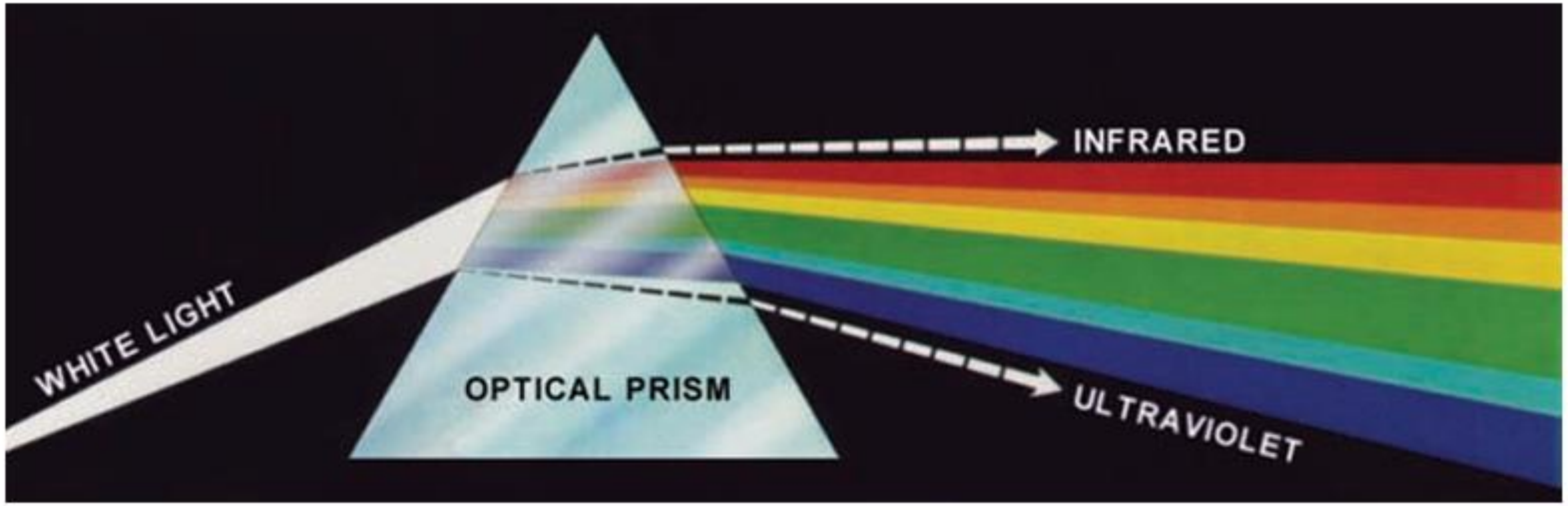
- Physical phenomenon
  - Physical nature of colour is known

- Psysio-psychological phenomenon
  - How human brain perceive and interpret colour?

colour fundamentals (cont.)

- 1666, Isaac Newton

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



Visible light

- Chromatic light span the electromagnetic spectrum (EM) from 400 to 700 nm

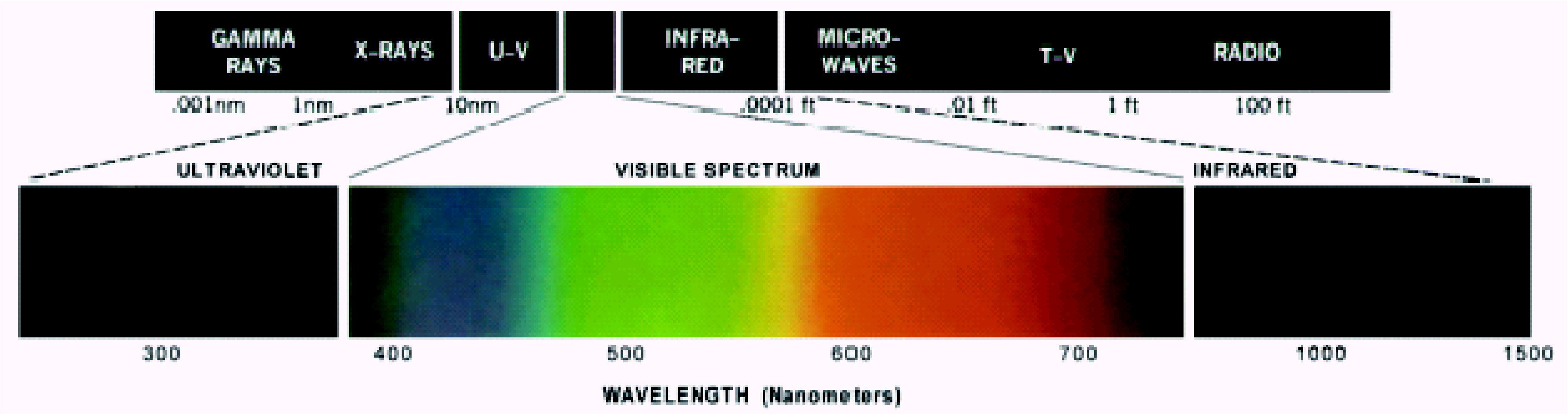
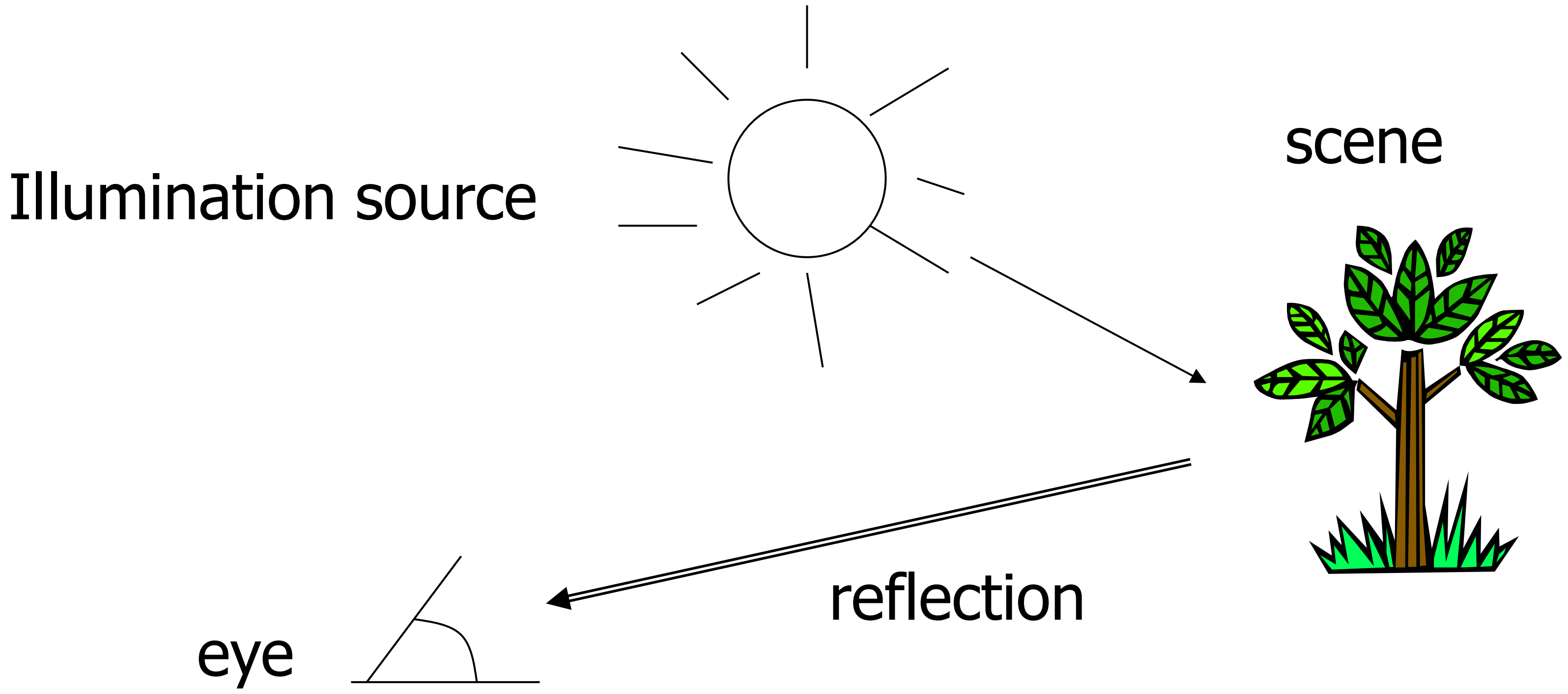


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

colour fundamentals (cont.)

- The colour that human perceive in an object = the light reflected from the object



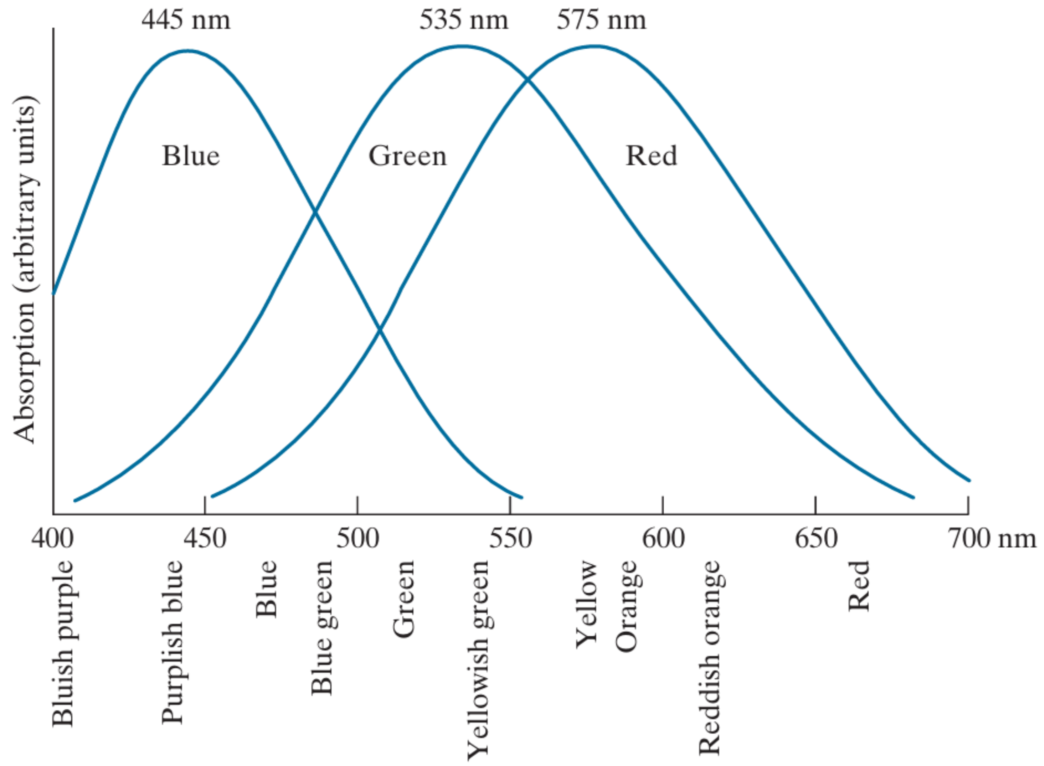
Physical quantities to describe a chromatic light source

- Radiance: total amount of energy that flow from the light source, measured in watts (W)
- Luminance: amount of energy an observer *perceives* from a light source, measured in lumens
  - Far infrared light: high radiance, but 0 luminance
- Brightness: subjective descriptor that is hard to measure, similar to the achromatic notion of intensity

How human eyes sense light?

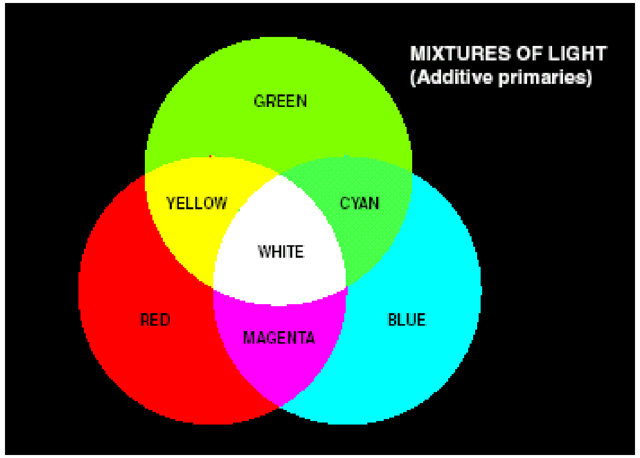
- 6~7M Cones are the sensors in the eye
- 3 principal sensing categories in eyes
  - Red light 65%,
  - green light 33%,
  - blue light 2%

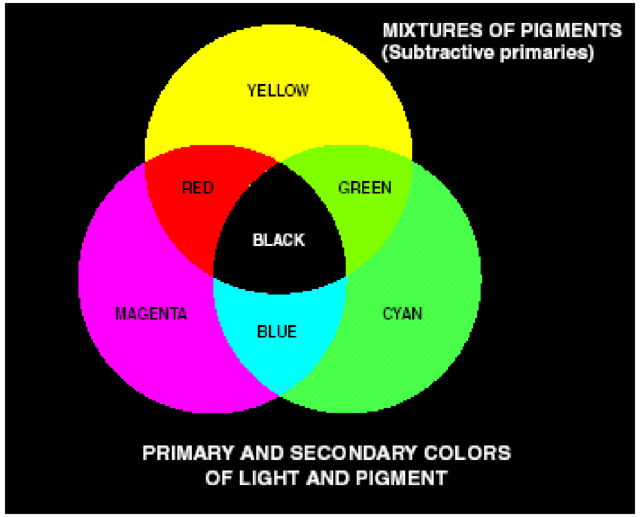
FIGURE 6.3
Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.



Primary and secondary colours

- In 1931, CIE(International Commission on Illumination) defines specific wavelength values to the primary colours
  - B = 435.8 nm, G = 546.1 nm, R = 700 nm
  - However, we know that no single colour may be called red, green, or blue
- Secondary colours: G+B=Cyan, R+G=Yellow, R+B=Magenta





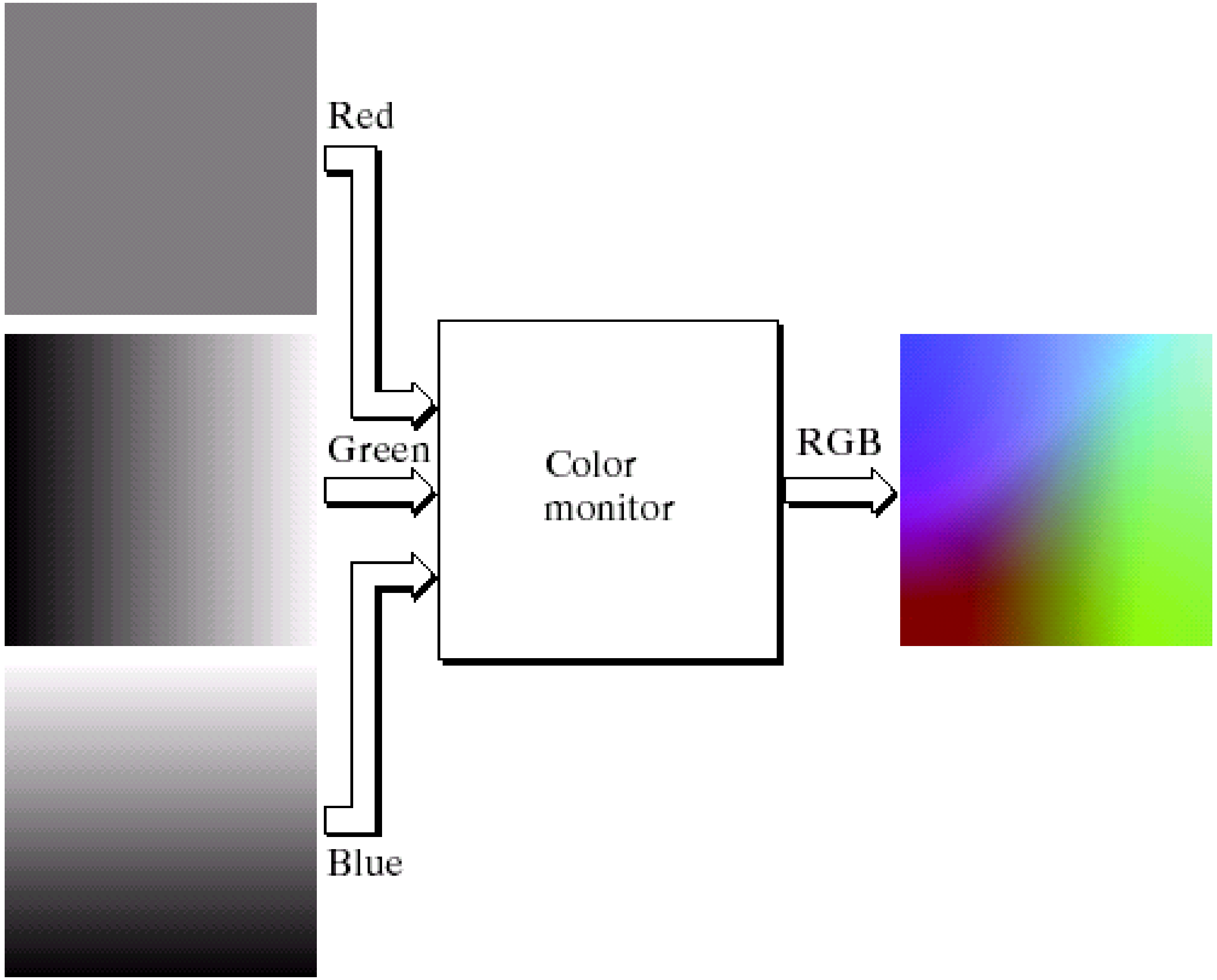
Primary colours of light v.s. primary colours of pigments

- Primary colour of pigments
  - colour that subtracts or absorbs a primary colour of light and reflects or transmits the other two

colour of light:	R	G	B
colour of pigments:	absorb R Cyan	absorb G Magenta	absorb B Yellow

Application of additive nature of light colours

- colour TV



CIE XYZ model

- RGB -> CIE XYZ model

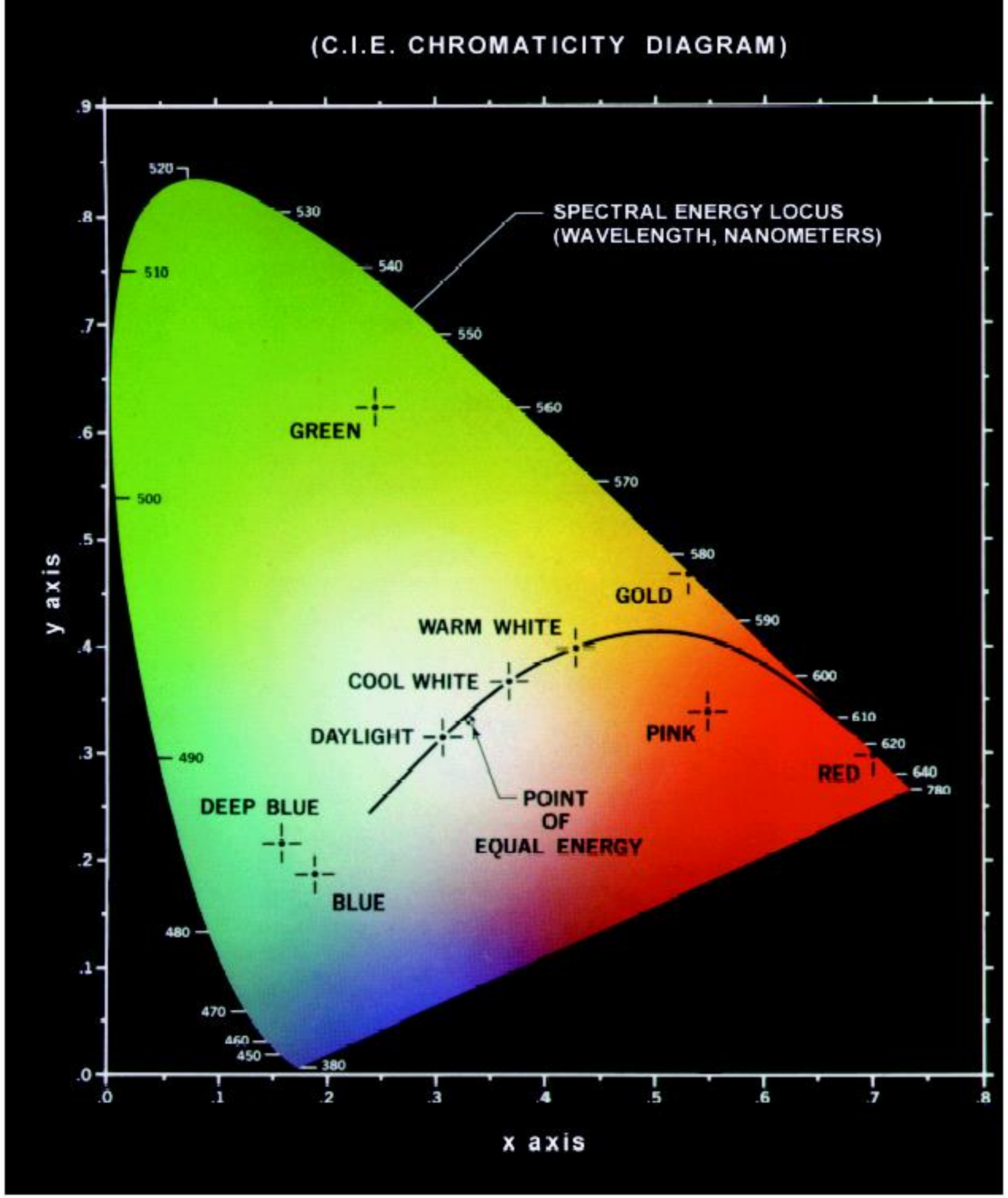
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Normalized tristimulus values

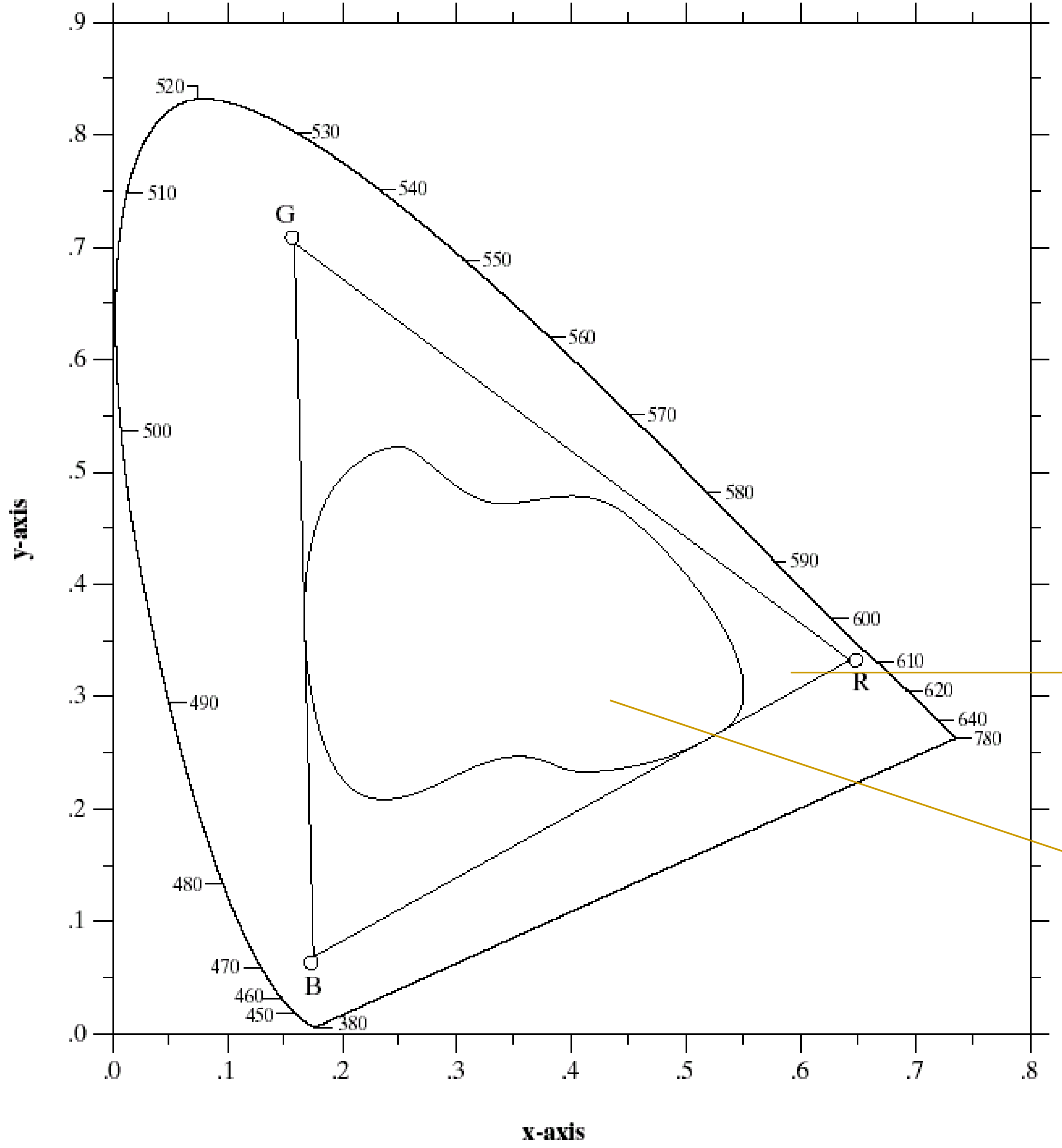
$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = \frac{Z}{X+Y+Z}$$

=> x+y+z=1. Thus, x, y (chromaticity coordinate) is enough to describe all colours

色度圖


(C.I.E. CHROMATICITY DIAGRAM)
SPECTRAL ENERGY LOCUS
(WAVELENGTH, NANOMETERS)
GREEN
GOLD
WARM WHITE
COOL WHITE
DAYLIGHT
PINK
RED
DEEP BLUE
BLUE
POINT
OF
EQUAL ENERGY
520
530
510
540
550
560
570
500
580
590
600
610
620
640
780
490
480
470
460
450
380
.9
.8
.7
.6
.5
.4
.3
.2
.1
.0
.0
.1
.2
.3
.4
.5
.6
.7
.8
y axis
x axis

By additivity of colours: Any colour inside the triangle can be produced by combinations of the three initial colours



RGB gamut of monitors

colour gamut of printers

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

colour models

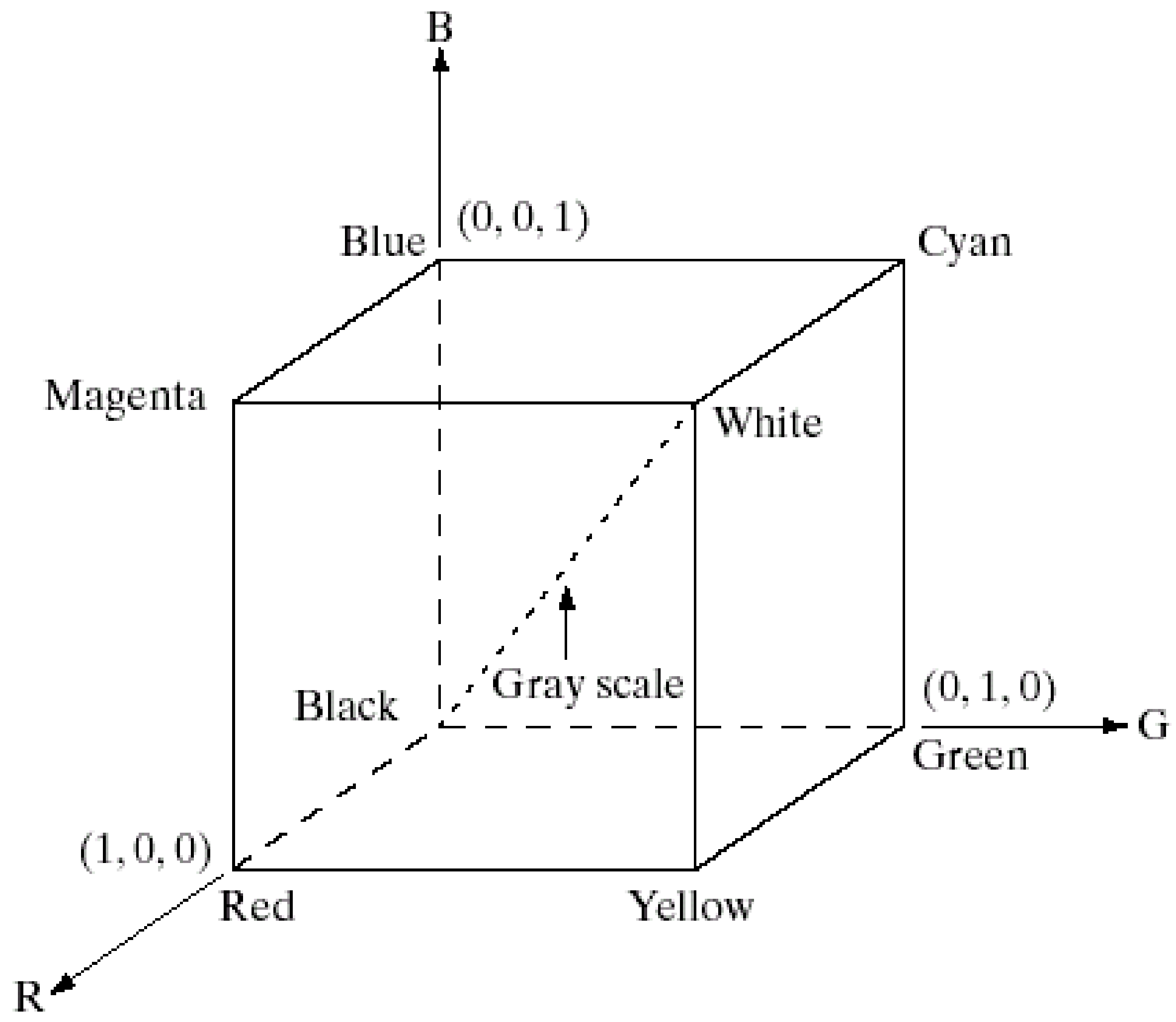
- colour model, colour space, colour system
  - Specify colours in a standard way
  - A coordinate system that each colour is represented by a single point

- RGB model
- CYM model
- CYMK model
- HSI model

Suitable for hardware or applications

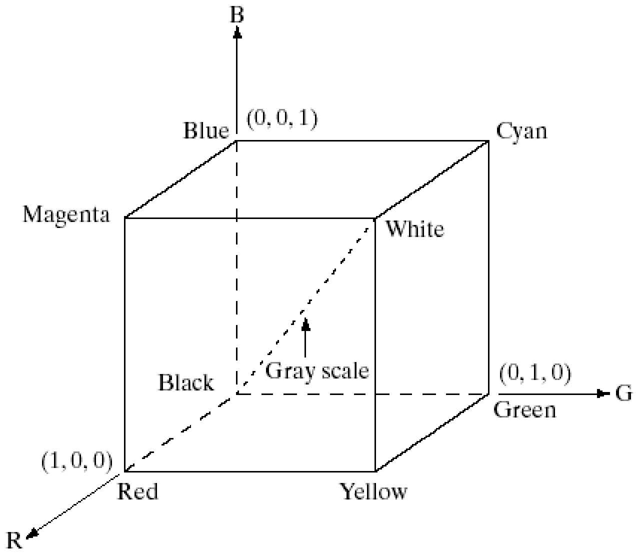
- match the human description

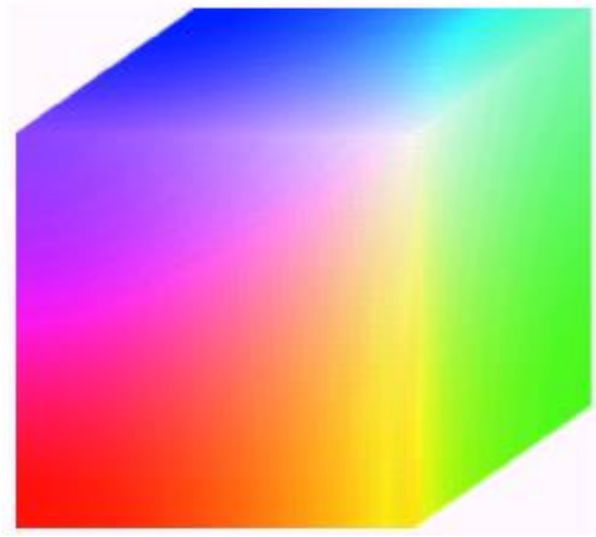
RGB colour model



Pixel depth

- Pixel depth: the number of bits used to represent each pixel in RGB space
- Full-colour image: 24-bit RGB colour image
  - (R, G, B) = (8 bits, 8 bits, 8 bits)





Safe RGB colours

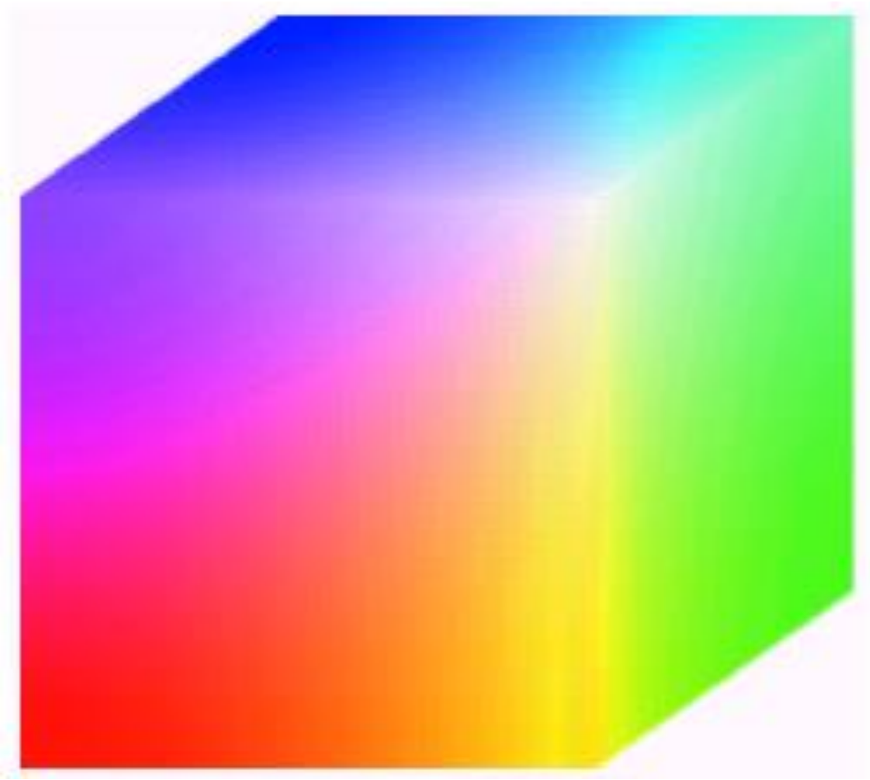
- Subset of colours is enough for some application
- Safe RGB colours (safe Web colours, safe browser colours)

Number System	Color Equivalents					
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255

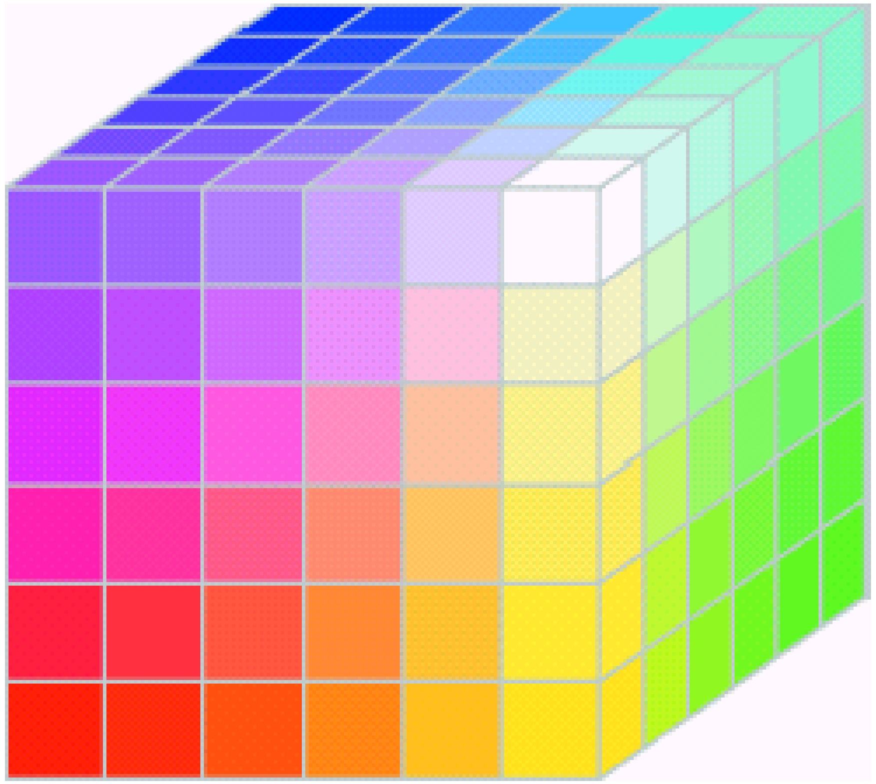
TABLE 6.1
Valid values of each RGB component in a safe color.

$(6)^3 = 216$

Safe RGB colour (cont.)



Full colour cube

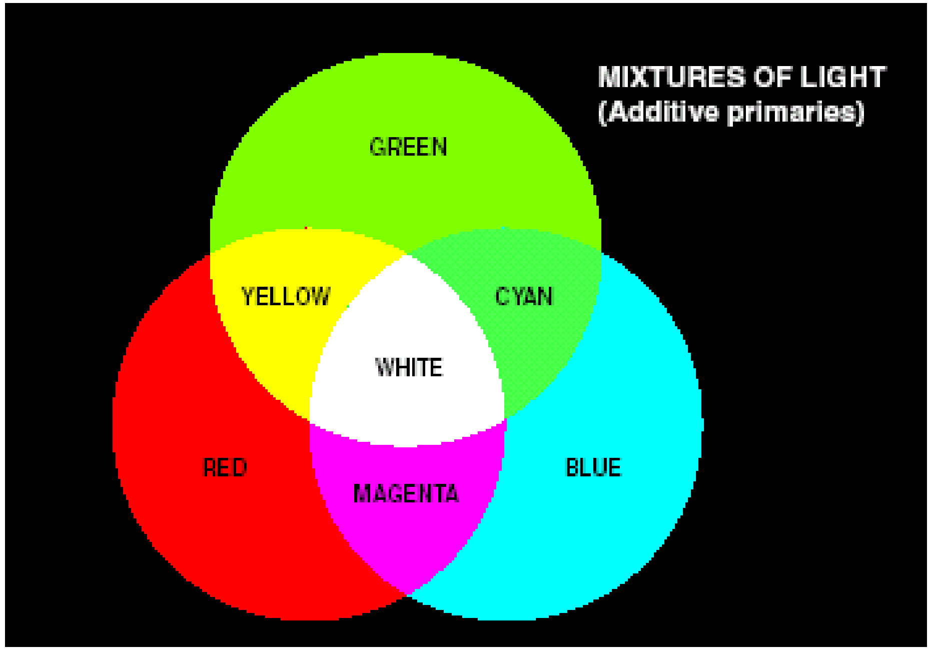


Safe colour cube

CMY model (+Black = CMYK)

- CMY: secondary colours of light, or primary colours of pigments
- Used to generate hardcopy output

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

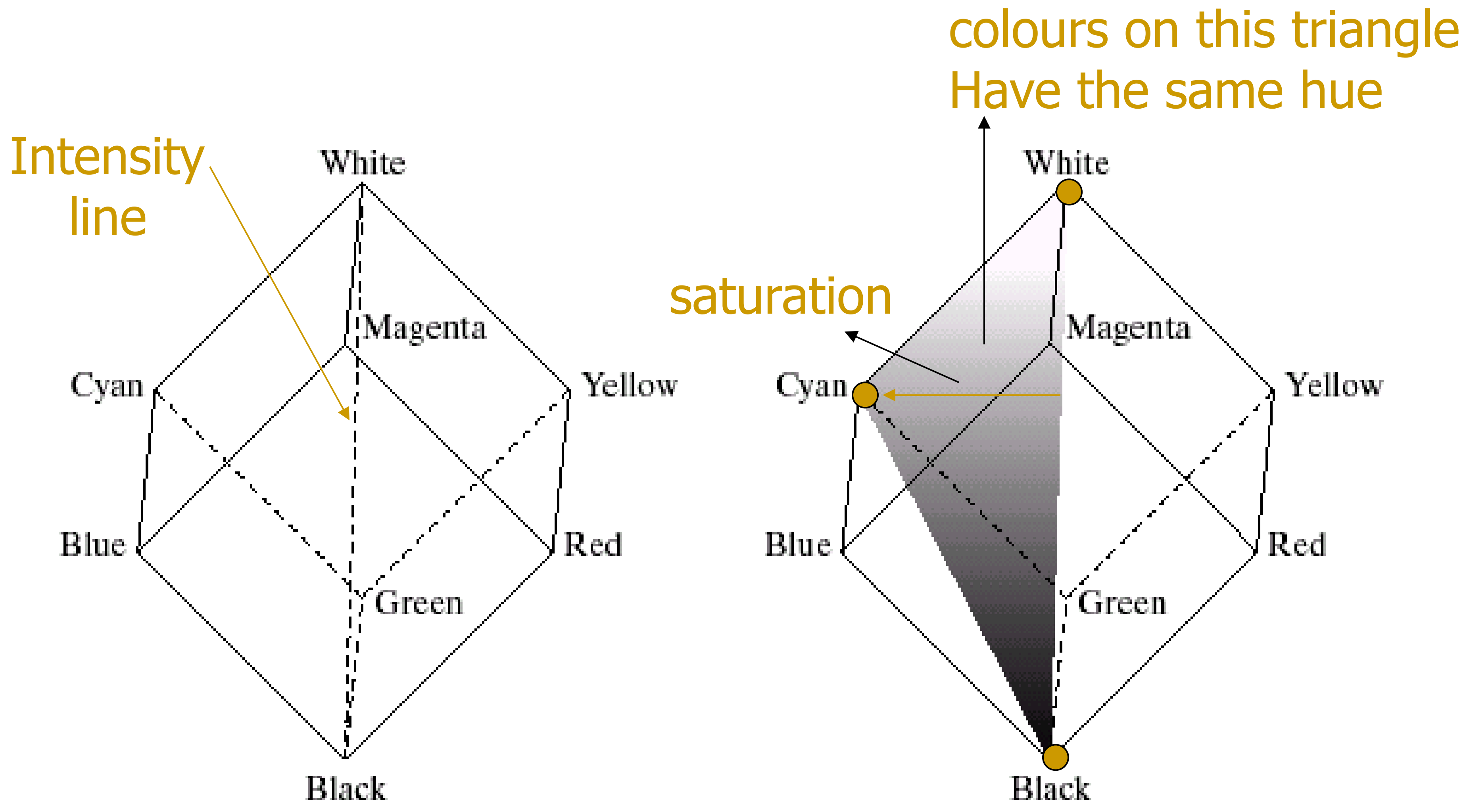


HSI colour model

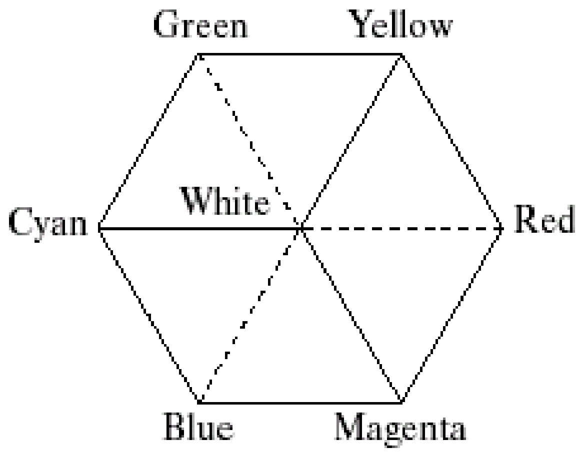
- Will you describe a colour using its R, G, B components?
- Human describe a colour by its hue, saturation, and brightness
  - Hue : colour attribute-> true or pure colour
  - Saturation: purity of colour (white->0, primary colour->1), mixtures of white colour with pure colours
  - Hue and saturation together is called Chromatic
  - Brightness: achromatic notion of intensity

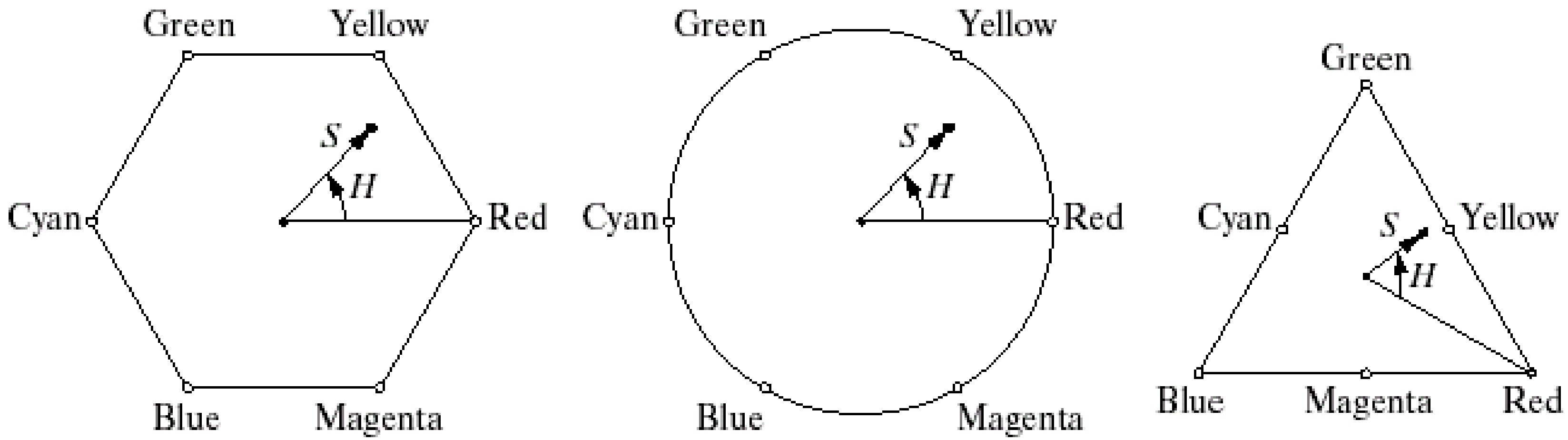
HSI colour model (cont.)

- RGB -> HSI model

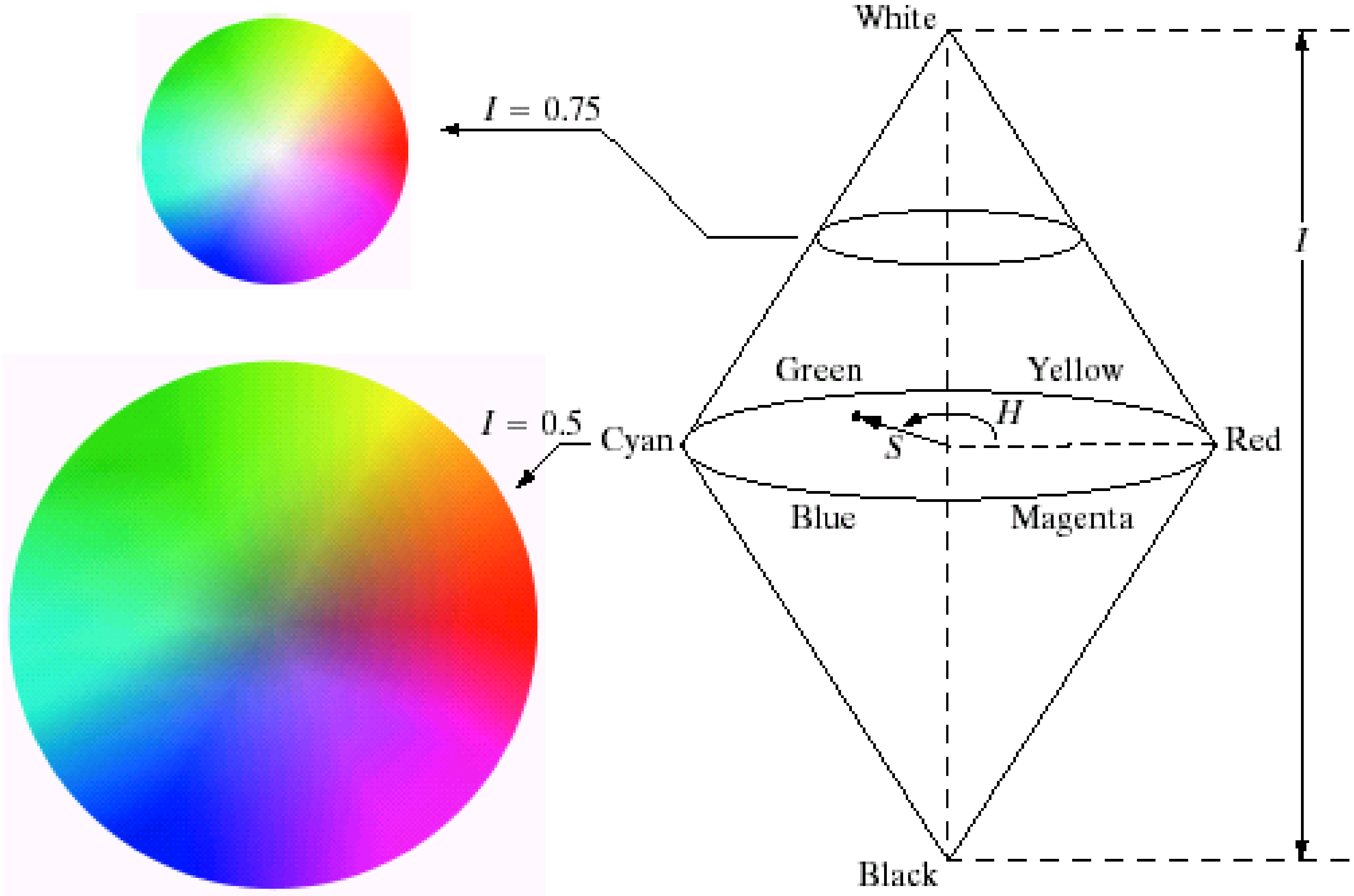


HSI model: hue and saturation





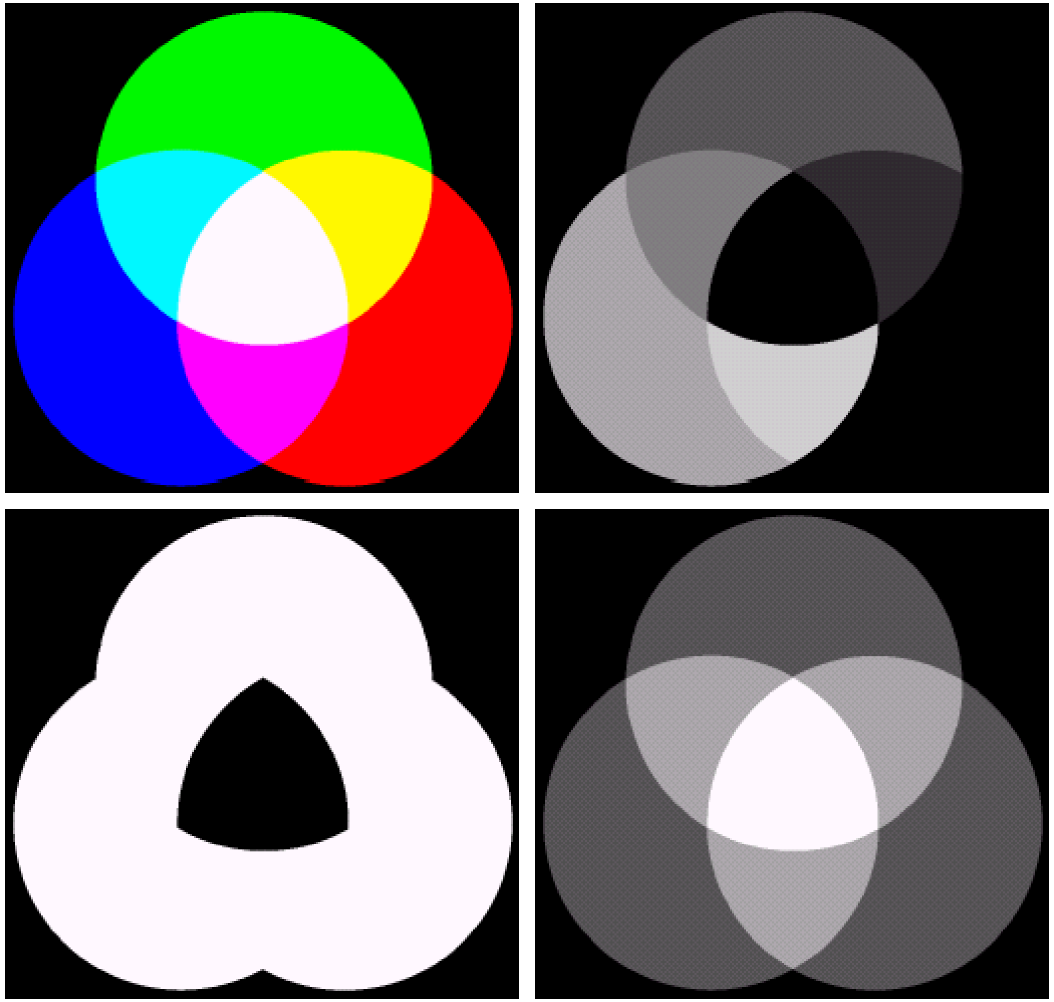
HSI model



HSI component images

R,G,B

Hue



saturation

intensity

Converting Colors from RGB to HSI

Given an image in RGB color format, the H component of each RGB pixel is obtained using the equation

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \tag{6-16}$$

with†

$$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \right\} \tag{6-17}$$

The saturation component is given by

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right] \tag{6-18}$$

Problem: Convert RGB (29, 104, 215) into HSI.

Converting Colors from HSI to RGB

RG sector $(0° \le H < 120°)$: When H is in this sector, the RGB components are given by the equations

$$B = I(1 - S) \quad (6\text{-}20)$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right] \quad (6\text{-}21)$$

and

$$G = 3I - (R + B) \quad (6\text{-}22)$$

GB sector $(120° \le H < 240°)$: If the given value of H is in this sector, we first subtract 120° from it:

$$H = H - 120° \quad (6\text{-}23)$$

Then, the RGB components are

$$R = I(1 - S) \quad (6\text{-}24)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right] \quad (6\text{-}25)$$

and

$$B = 3I - (R + G) \quad (6\text{-}26)$$

BR sector $(240° \le H \le 360°)$: Finally, if H is in this range, we subtract 240° from it:

$$H = H - 240° \quad (6\text{-}27)$$

Then, the RGB components are

$$G = I(1 - S) \quad (6\text{-}28)$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right] \quad (6\text{-}29)$$

and

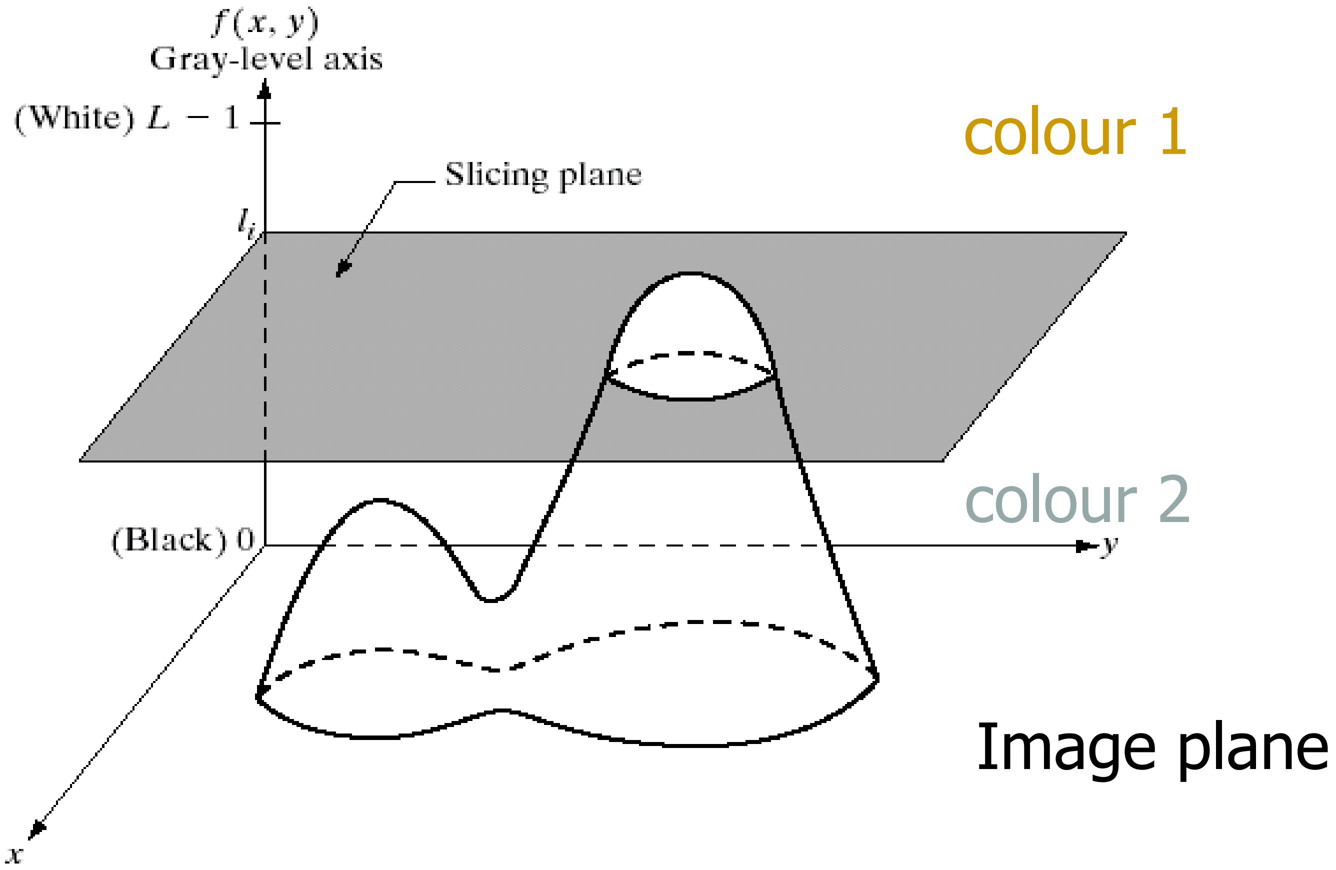
$$R = 3I - (G + B) \quad (6\text{-}30)$$

Pseudo-colour image processing

- Assign colours to gray values based on a specified criterion
- For human visualization and interpretation of gray-scale events
- Intensity slicing
- Gray level to colour transformations

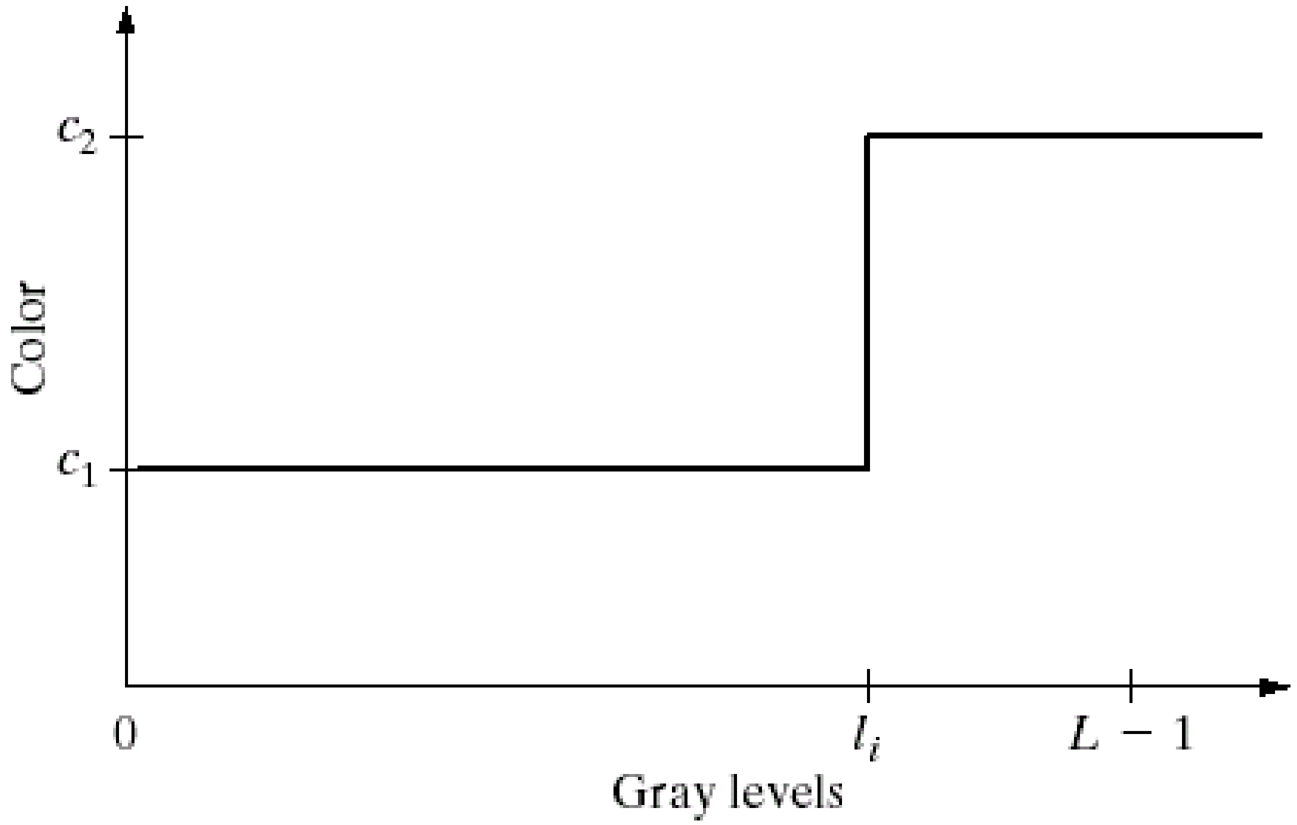
Intensity slicing

- 3-D view of intensity image



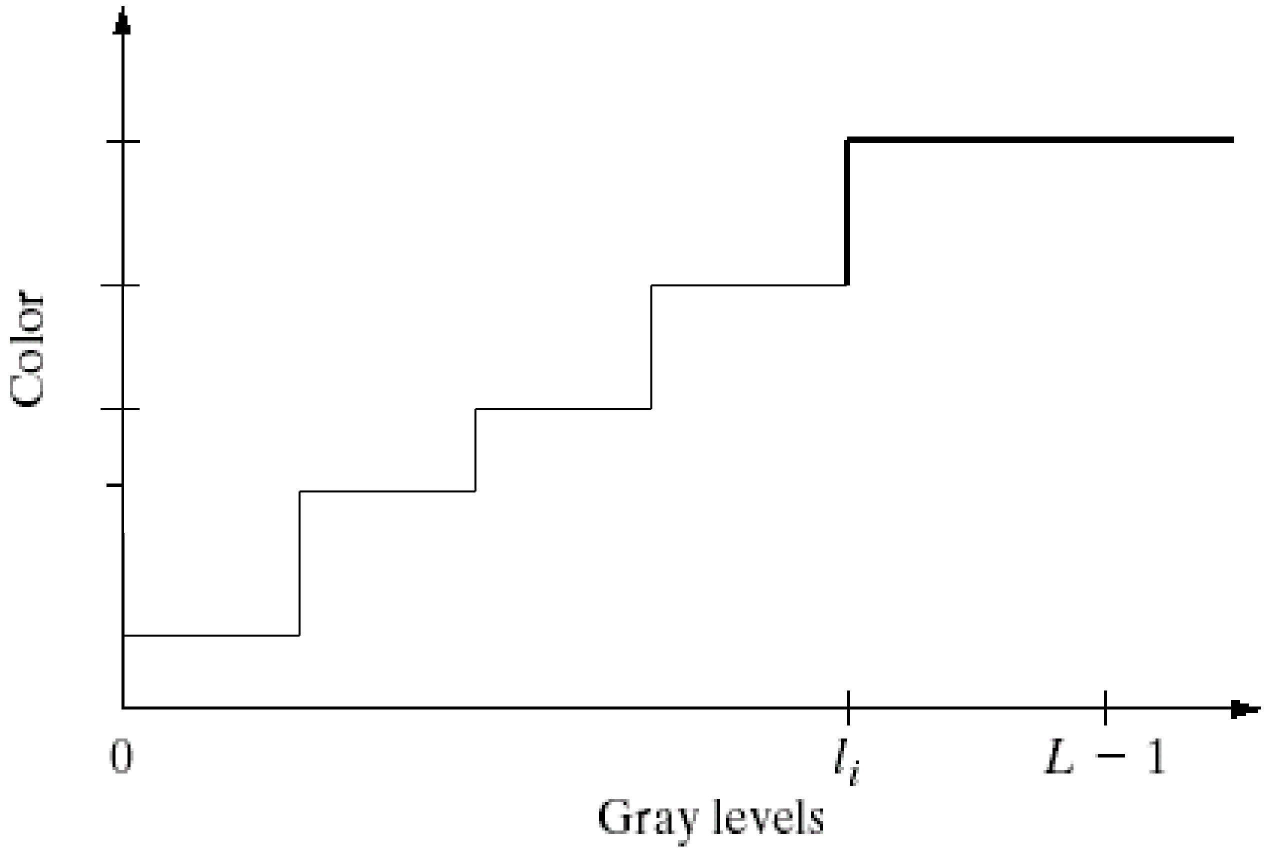
Intensity slicing (cont.)

- Alternative representation of intensity slicing

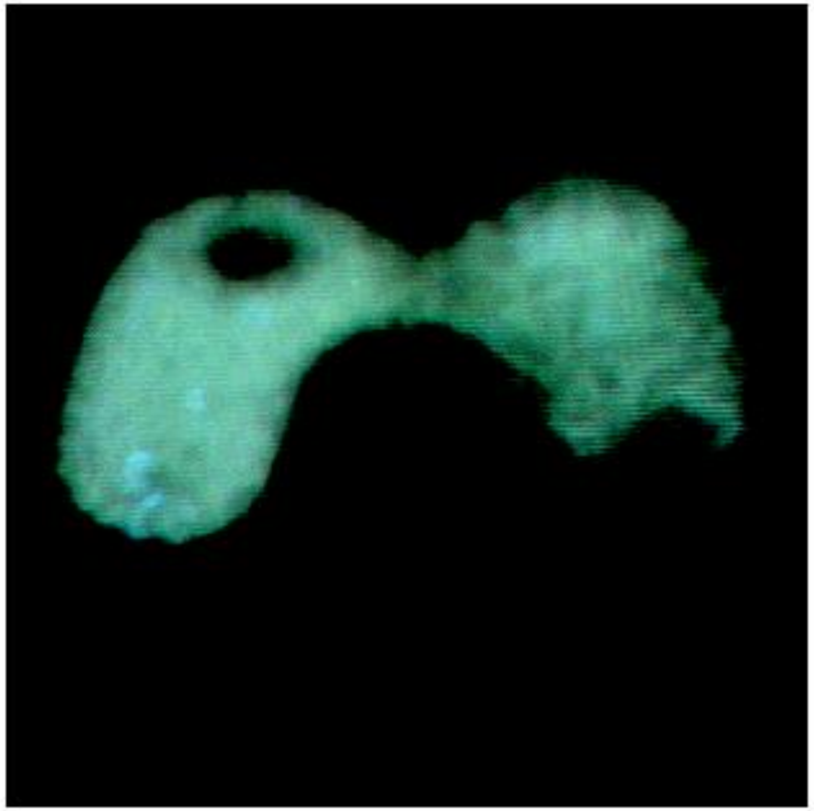


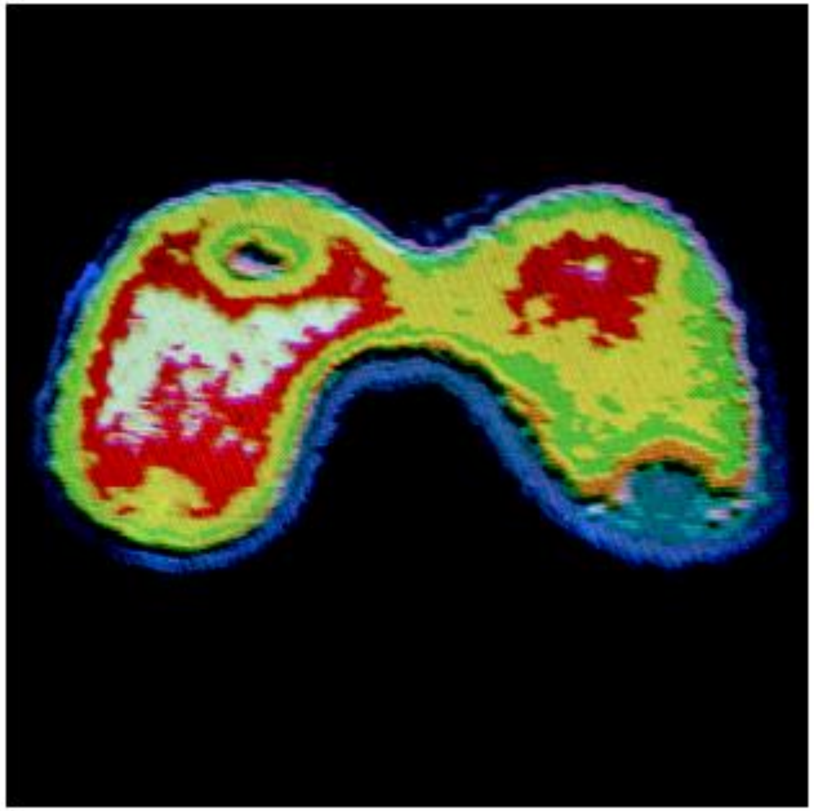
Intensity slicing (cont.)

- More slicing plane, more colours



Application 1

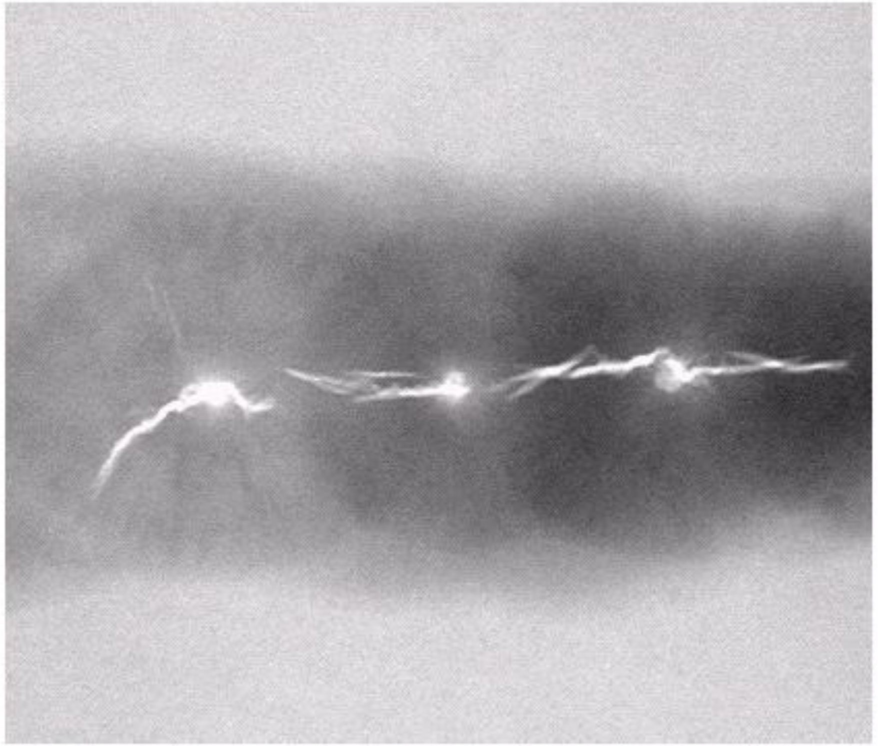




Radiation test pattern → 8 colour regions

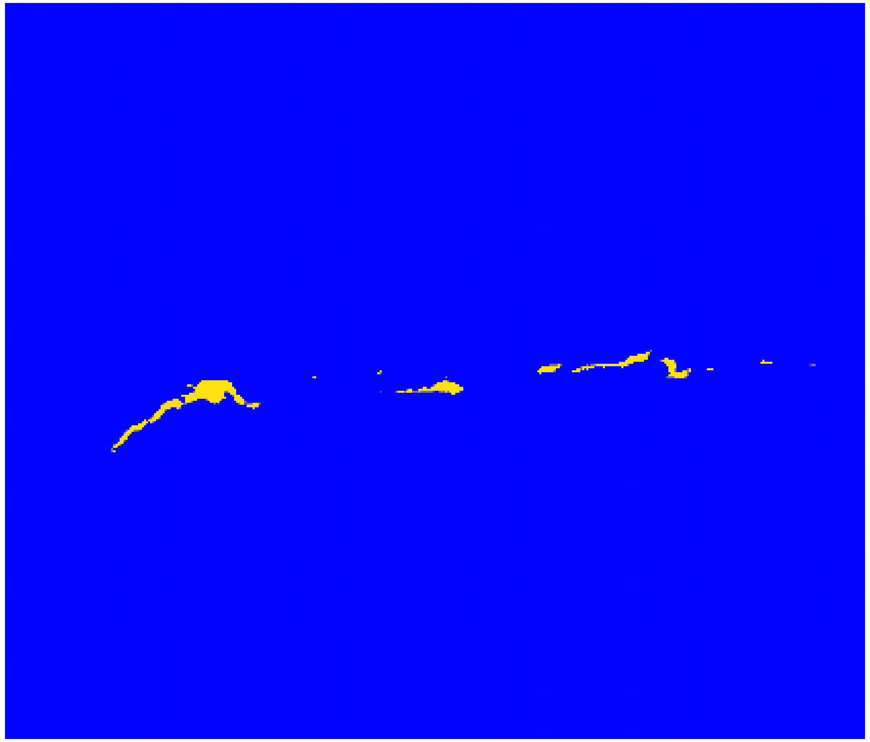
* See the gradual gray-level changes

Application 2

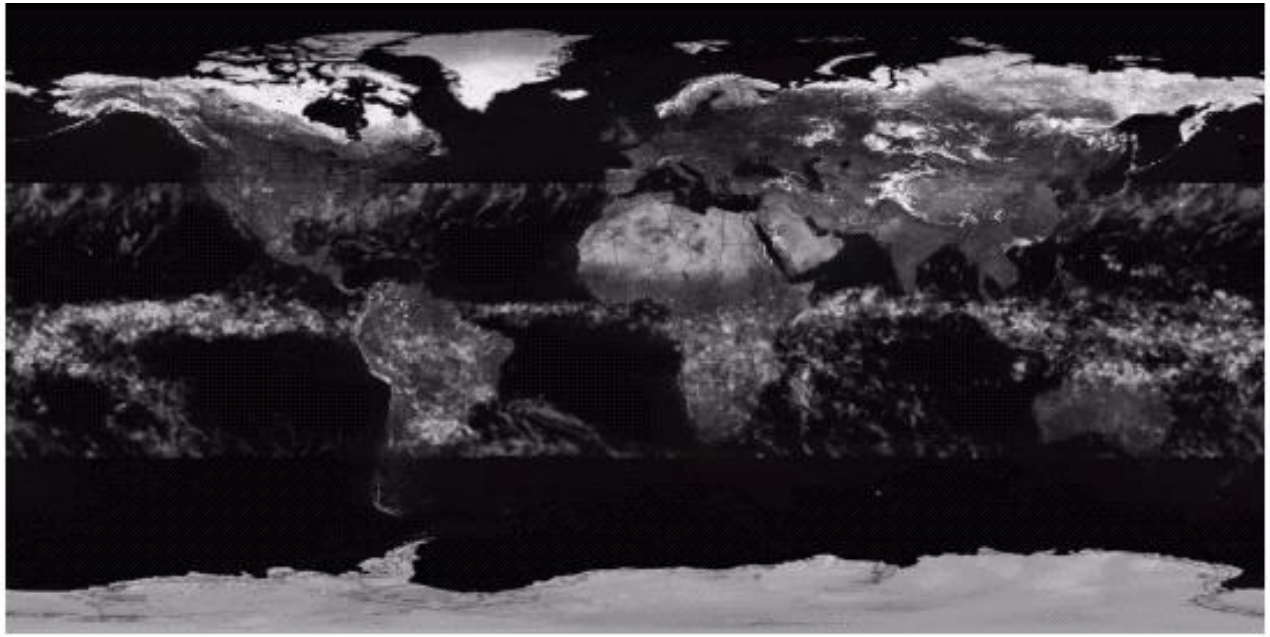


X-ray image of a weld

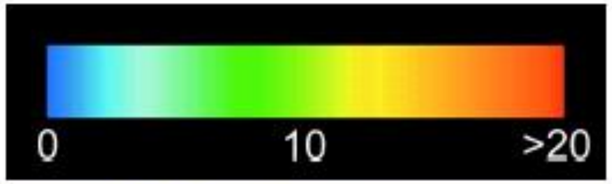
焊接物

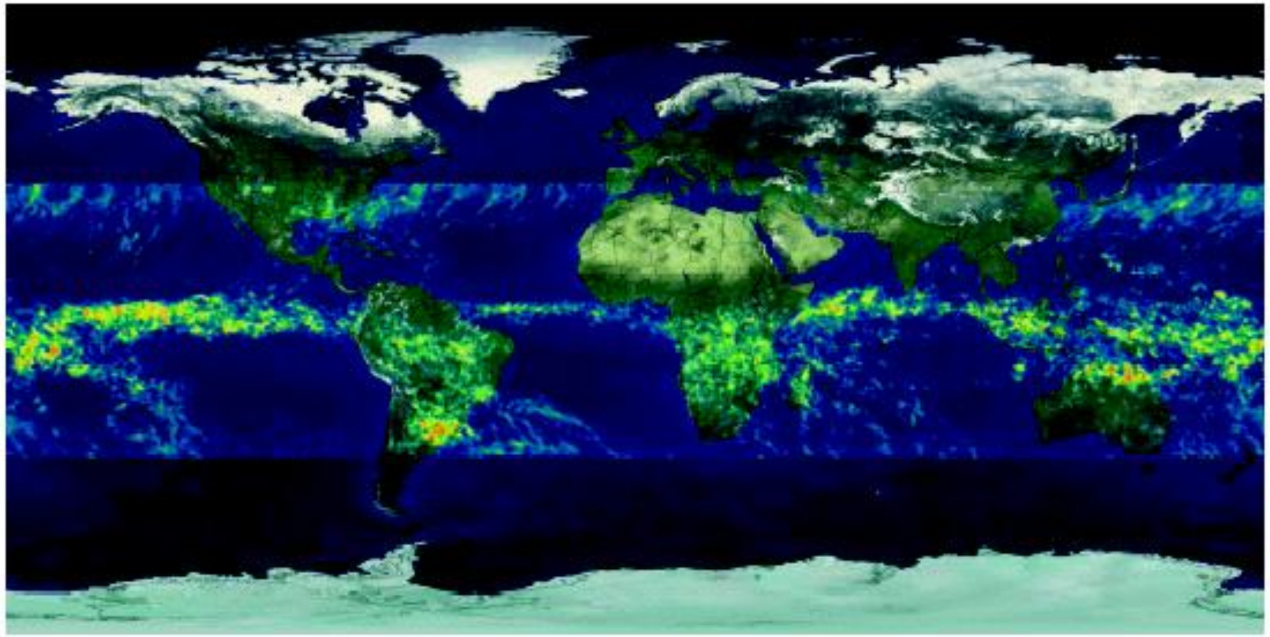


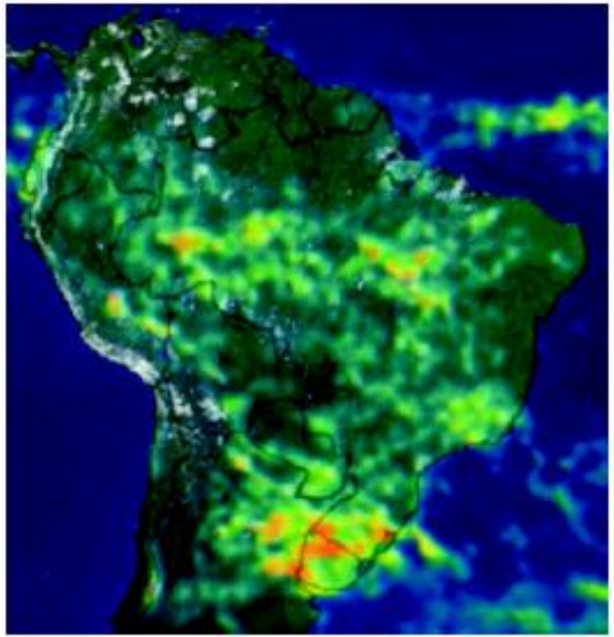
Application 3



Rainfall statistics

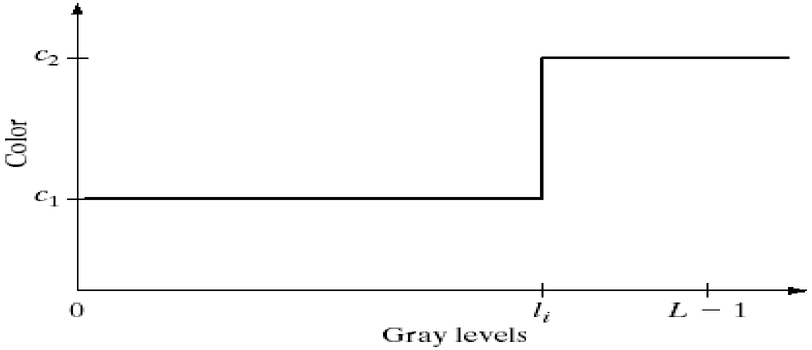




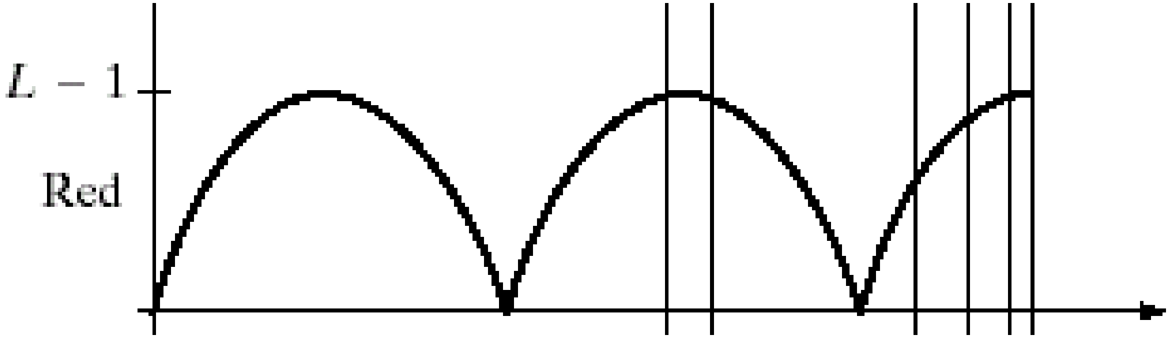


Gray level to colour transformation

- Intensity slicing: piecewise linear transformation



- ormation



Gray level to colour transformation

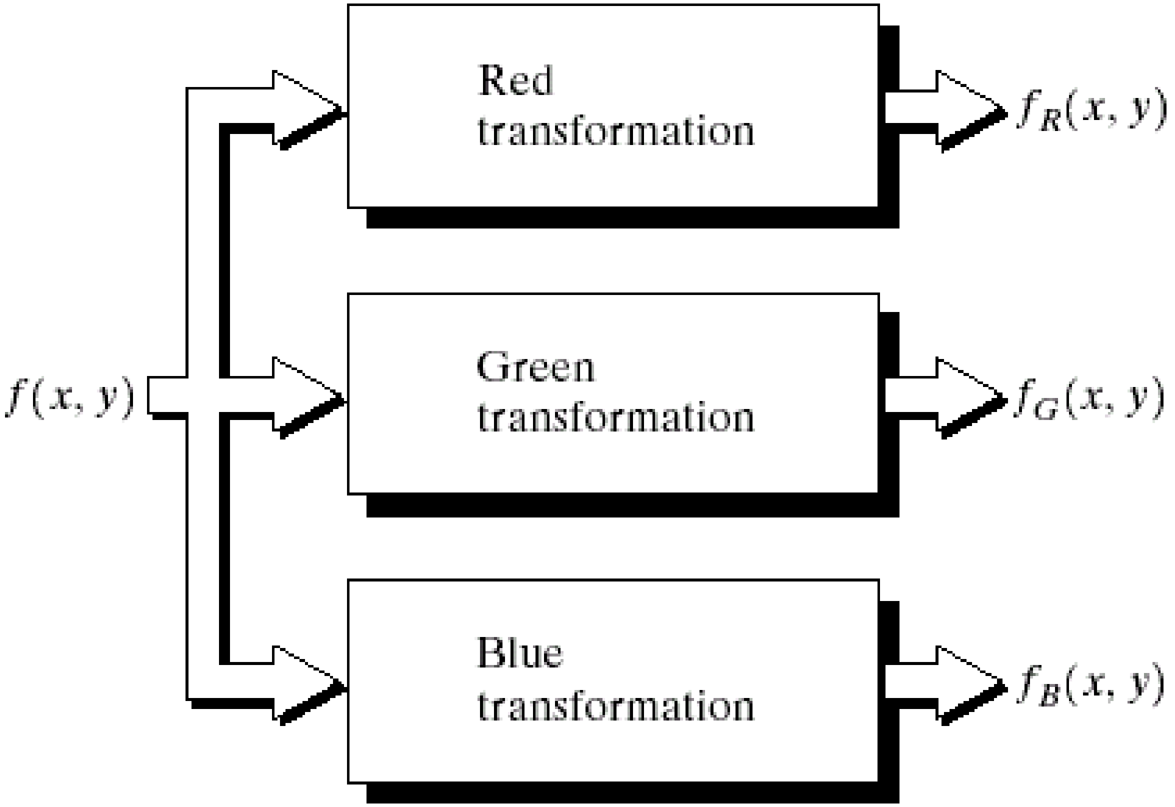
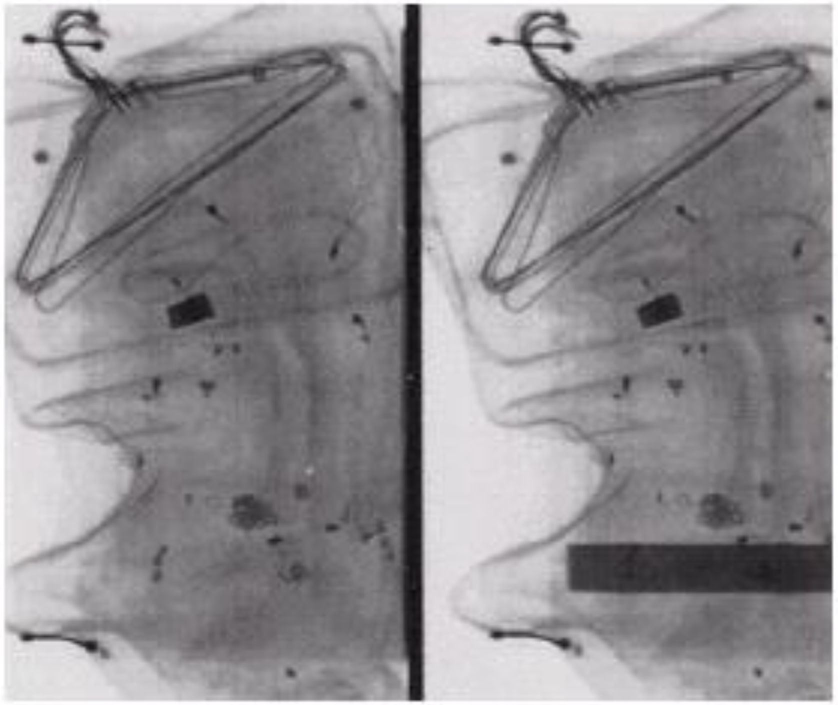


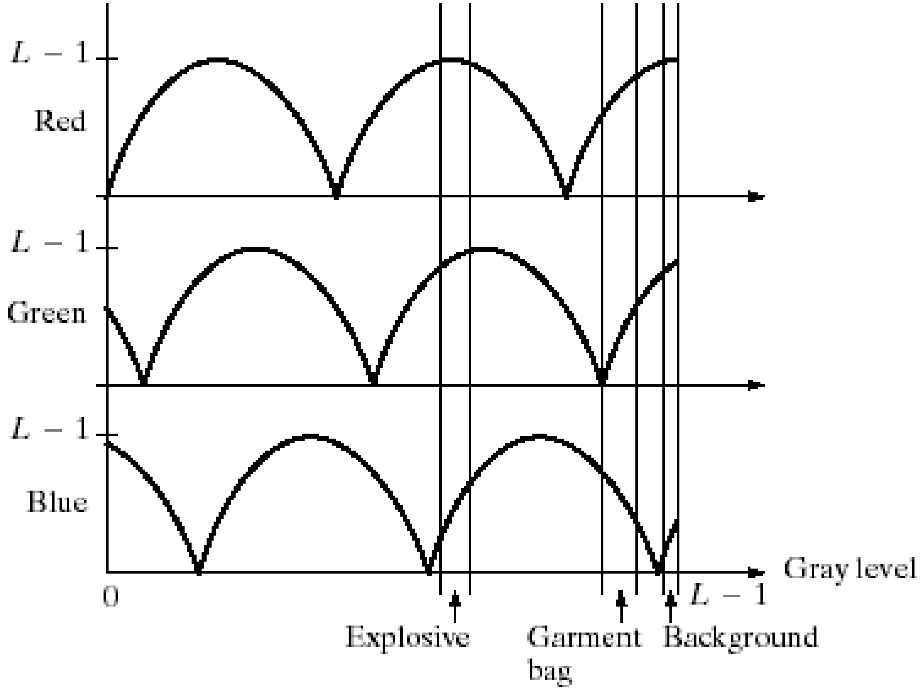
FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

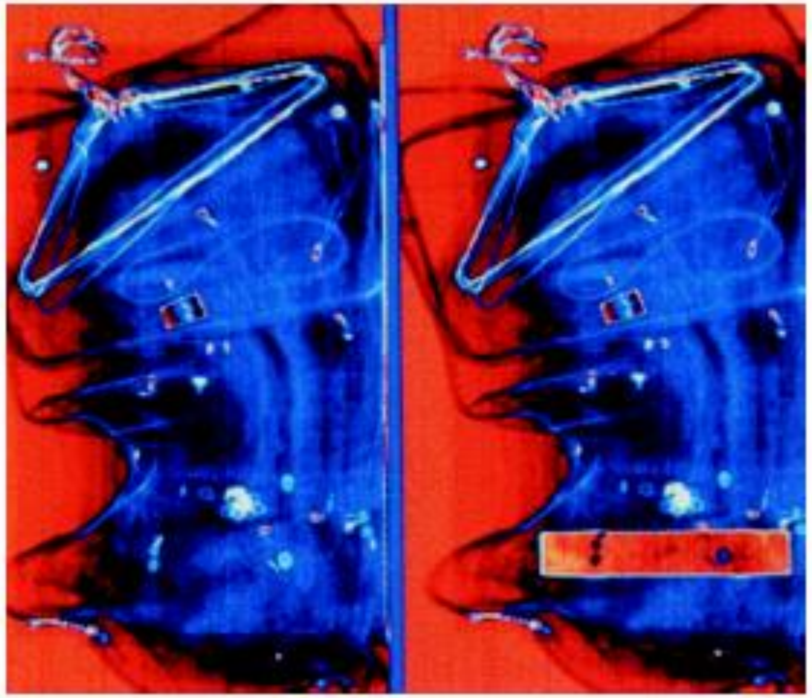
Gray level to Colour Transformation

- Gray level colour assignment can be done according to the following relation:
  - $f(x, y) = c_k$ if $f(x, y) \in v_k$
  - Where c_k is the colour assigned to the kth intensity level v_k defined by the partitioning plane at l=k-1 and l=k ; => $f(x, y) = c_1$ if $f(x, y) \in v_1$
  - If we want green component of any image by gray level colour transformation we can apply
  - $f_R(x, y) = 0.33f(x, y)$
  - $f_G(x, y) = f(x, y)$
  - $f_B(x, y) = 0.11f(x, y)$

Application 1

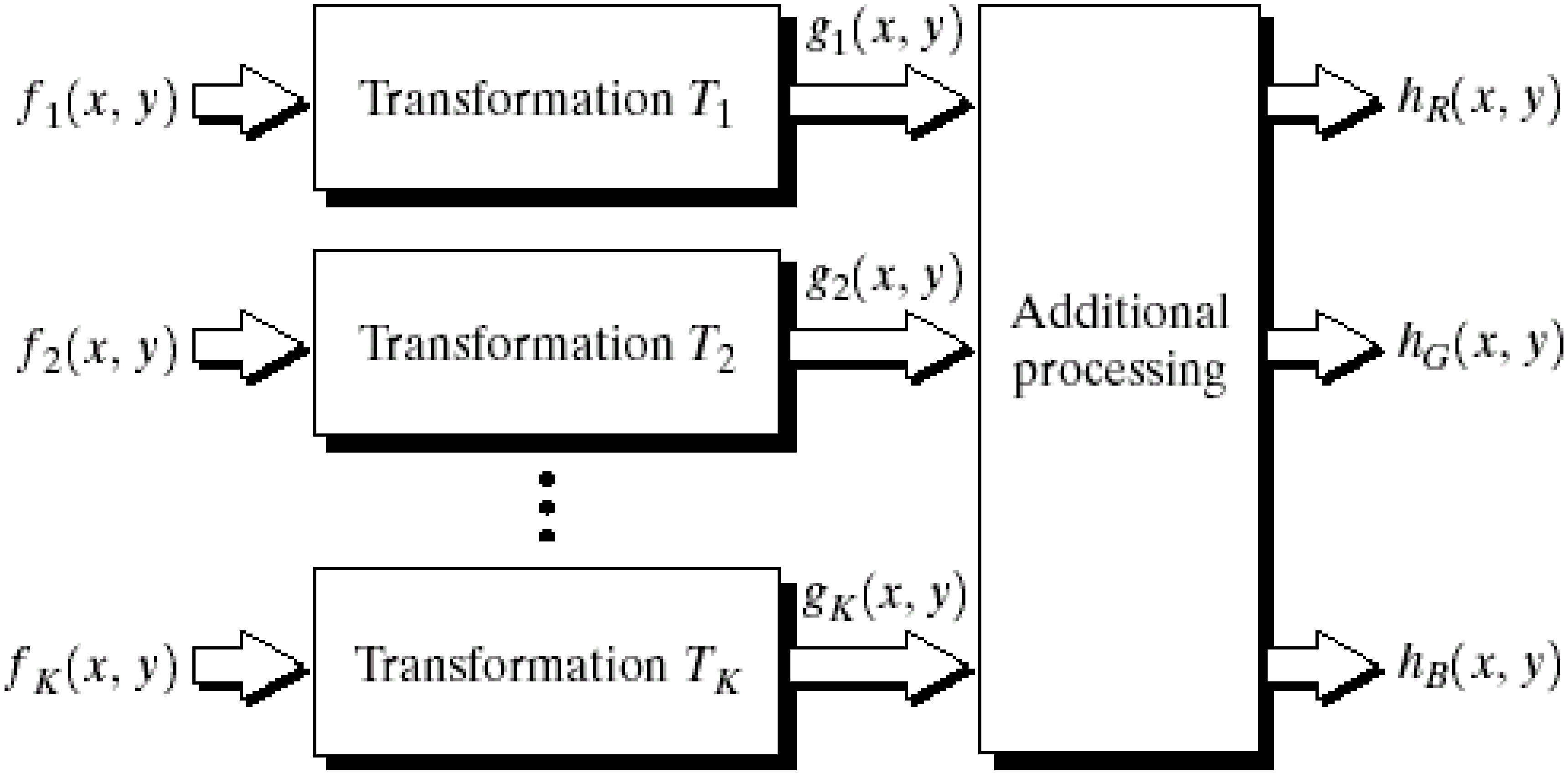






Combine several monochrome images

Example: multi-spectral images



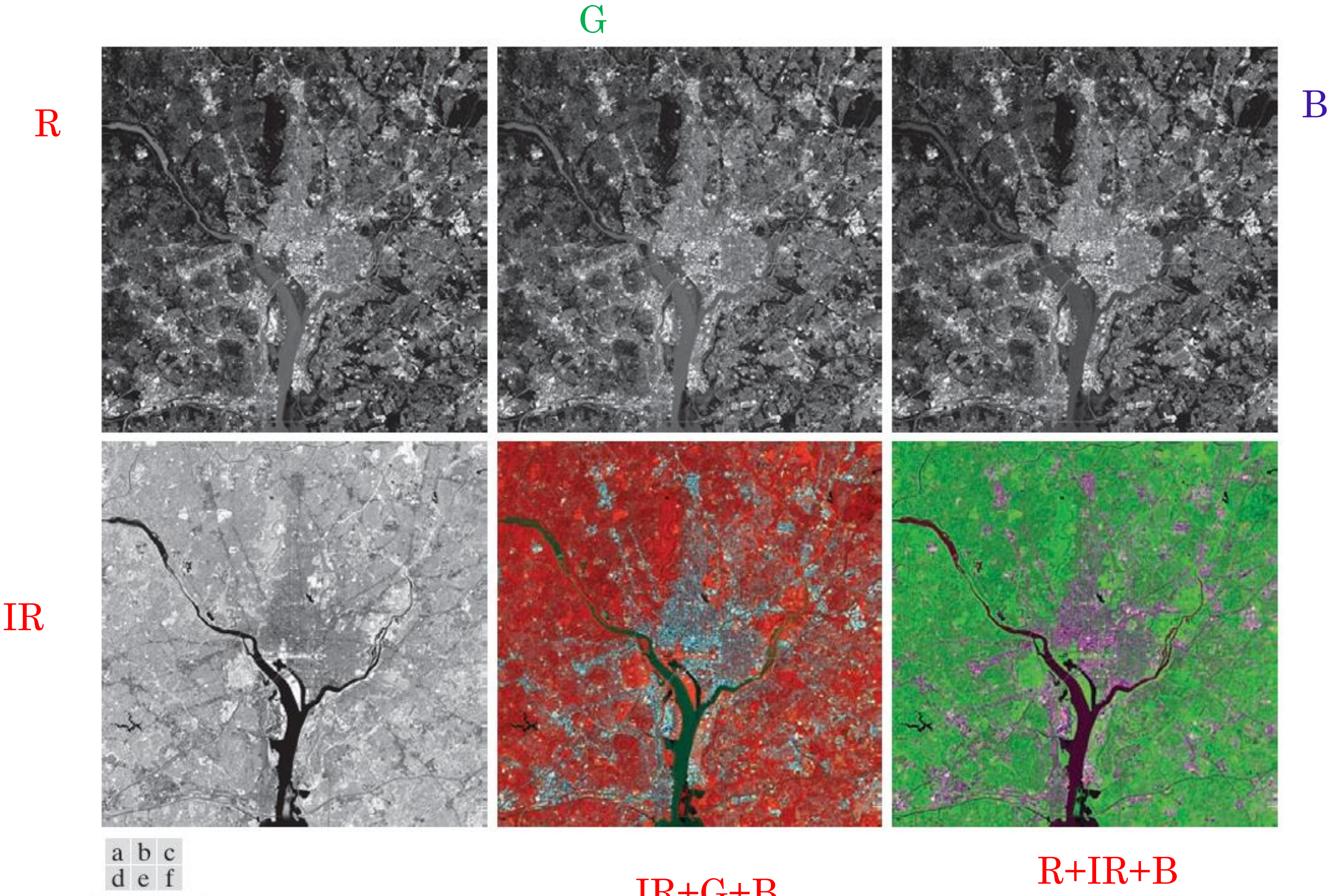


FIGURE 6.25 (a)–(d) Red (R), green (G), blue (B), and near-infrared (IR) components of a LANDSAT multispectral image of the Washington, D.C. area. (e) RGB color composite image obtained using the IR, G, and B component images. (f) RGB color composite image obtained using the R, IR, and B component images. (Original multispectral images courtesy of NASA.)

Colour Pixel

- A pixel at (x,y) is a vector in the colour space
  - RGB colour space

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

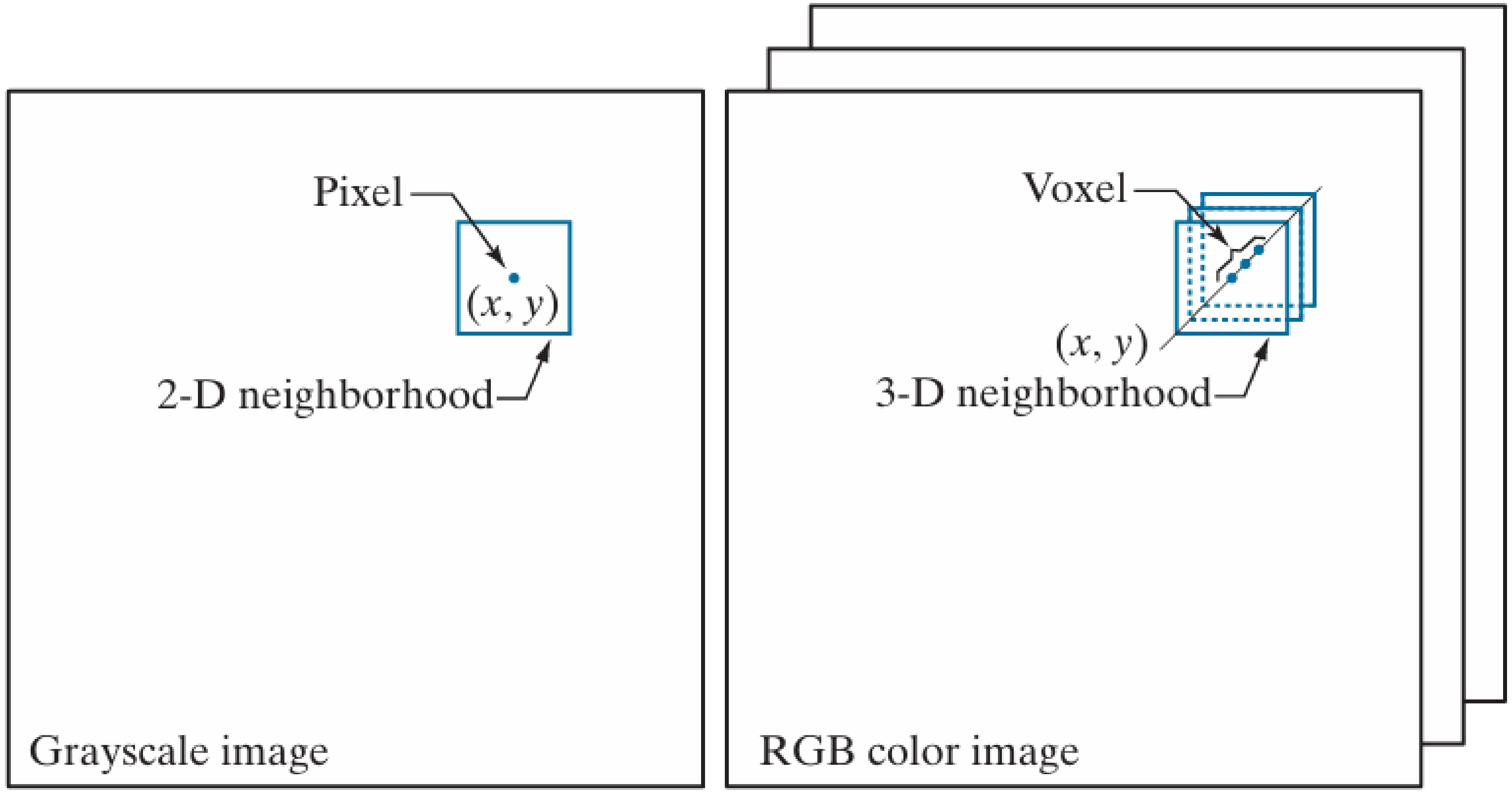
c.f. gray-scale image

f(x,y) = I(x,y)

Example: Spatial Mask

a b

FIGURE 6.27
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.



How to deal with colour vector?

- Per-colour-component processing
  - Process each colour component
- Vector-based processing
  - Process the colour vector of each pixel
- When can the above methods be equivalent?
  - Process can be applied to both scalars and vectors
  - Operation on each component of a vector must be independent of the other component

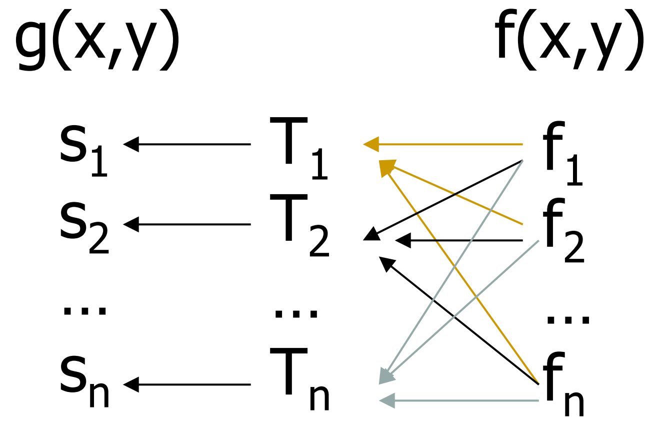
Two Spatial Processing Categories

- Similar to gray scale processing studied before, we have to major categories
- Pixel-wise processing
- Neighborhood processing

Colour Transformation

- Similar to gray scale transformation
  - g(x,y)=T[f(x,y)]
- Colour transformation

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



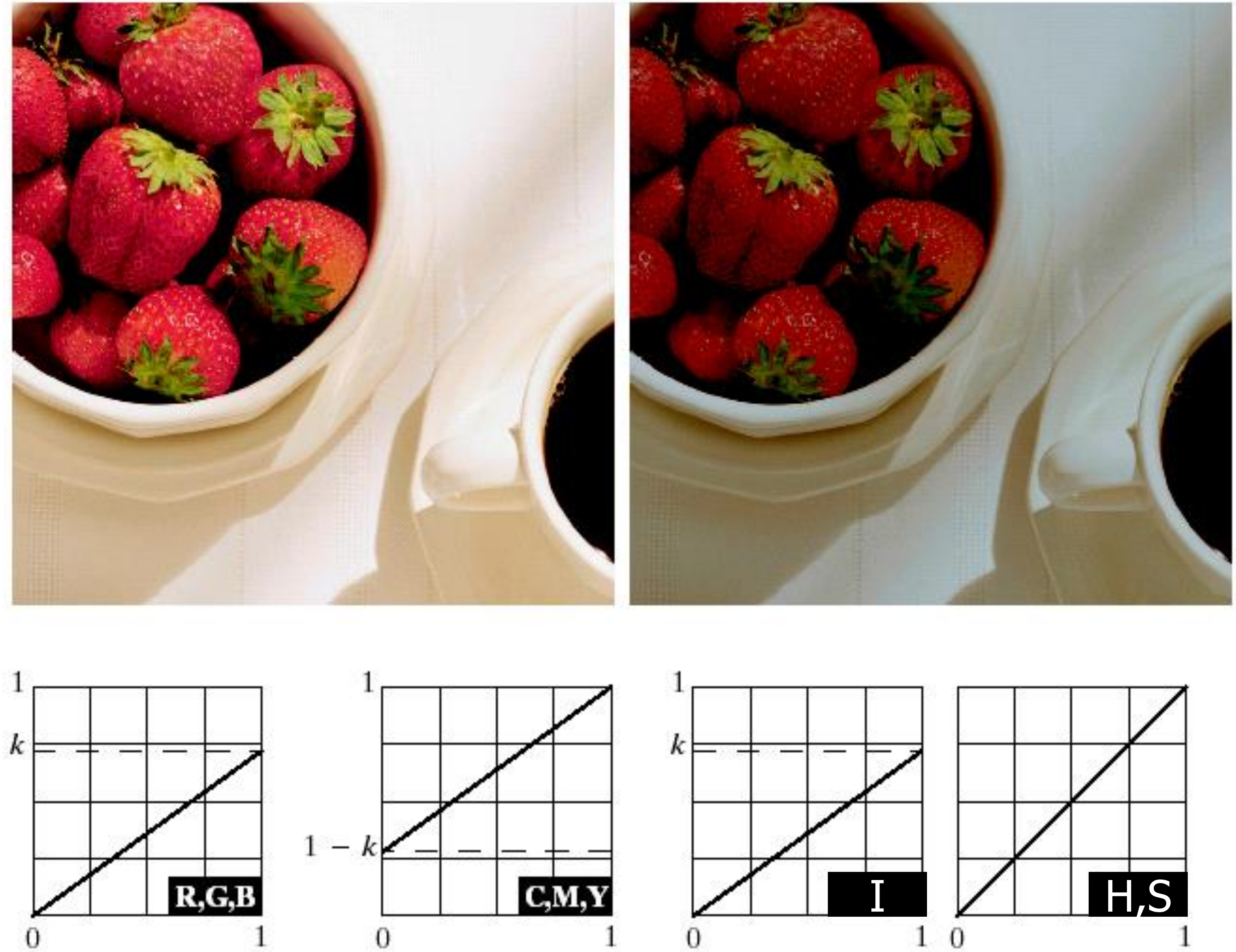
Use which colour model in colour transformation?

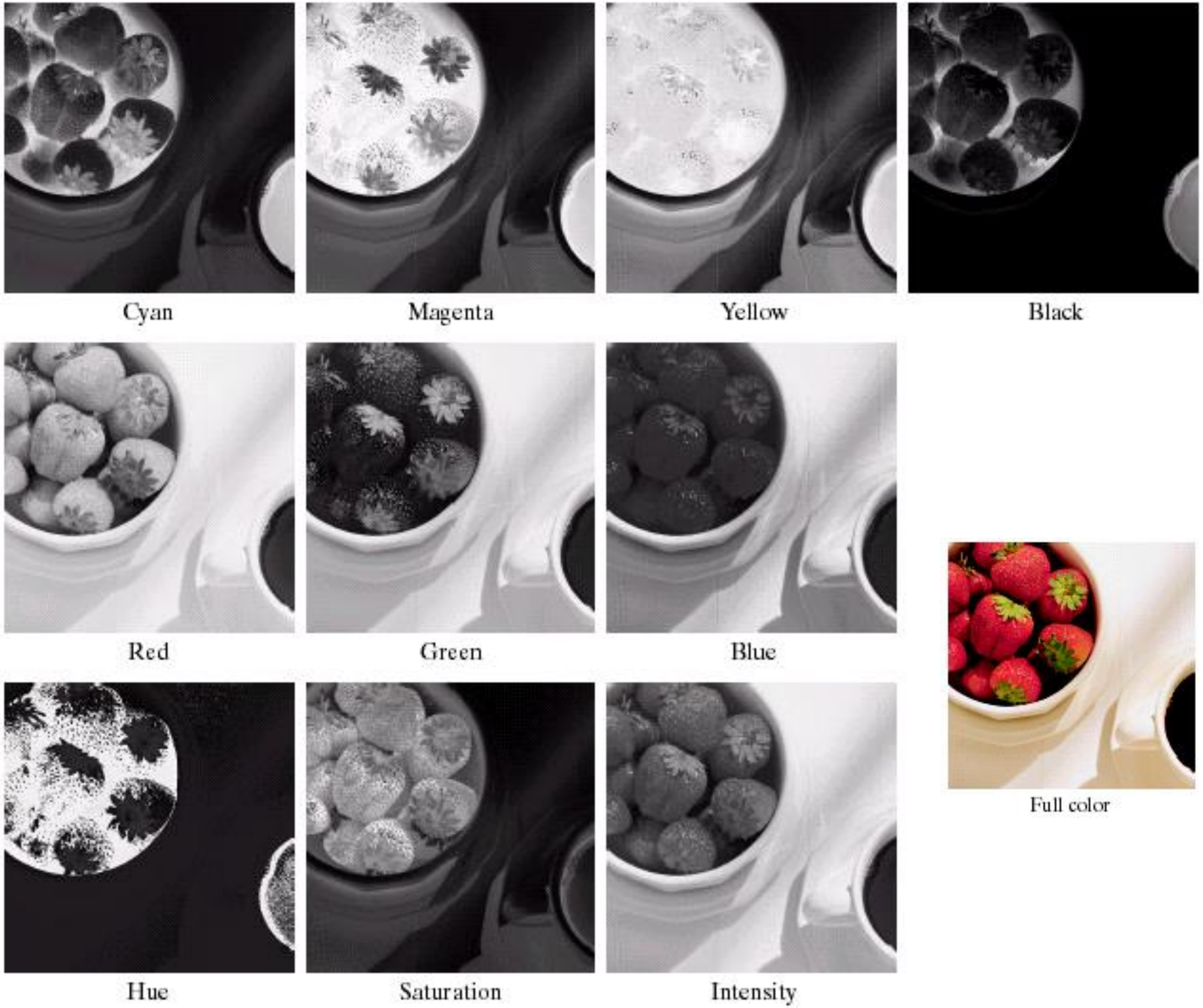
- RGB ⇔CMY(K) ⇔ HSI
- Theoretically, any transformation can be performed in any colour model
- Practically, some operations are better suited to specific colour model

Example: modify intensity of a colour image

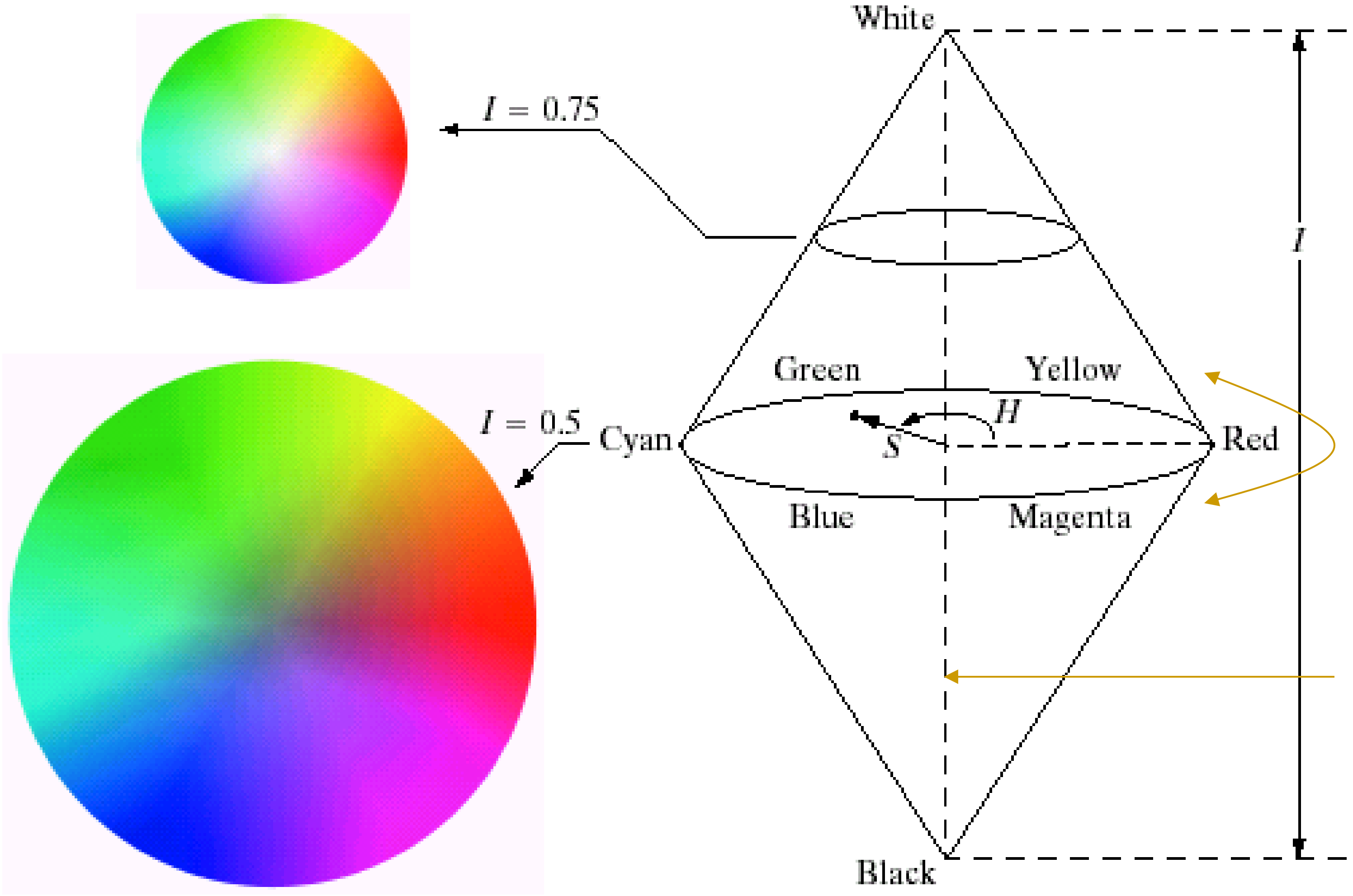
- **Example:** $g(x,y)=k\,f(x,y)$, $0<k<1$
- HSI colour space
  - Intensity: $s_3 = kr_3$
  - Note: transform to HSI requires complex operations
- RGB colour space
  - For each R,G,B component: $s_i = kr_i$; $i = 1,2,3 \ldots.n$
- CMY colour space
  - For each C,M,Y component:
  - $s_i = k\,r_i + (1-k)$
- CMYK colour space

$$s_i = \begin{cases} r_i & i = 1,2,3 \\ kr_i + (1-k) & i = 4 \end{cases}$$


R,G,B
C,M,Y
I
H,S
k
1 − k
0
1


Cyan
Magenta
Yellow
Black
Red
Green
Blue
Hue
Saturation
Intensity
Full color

Problem of using Hue component

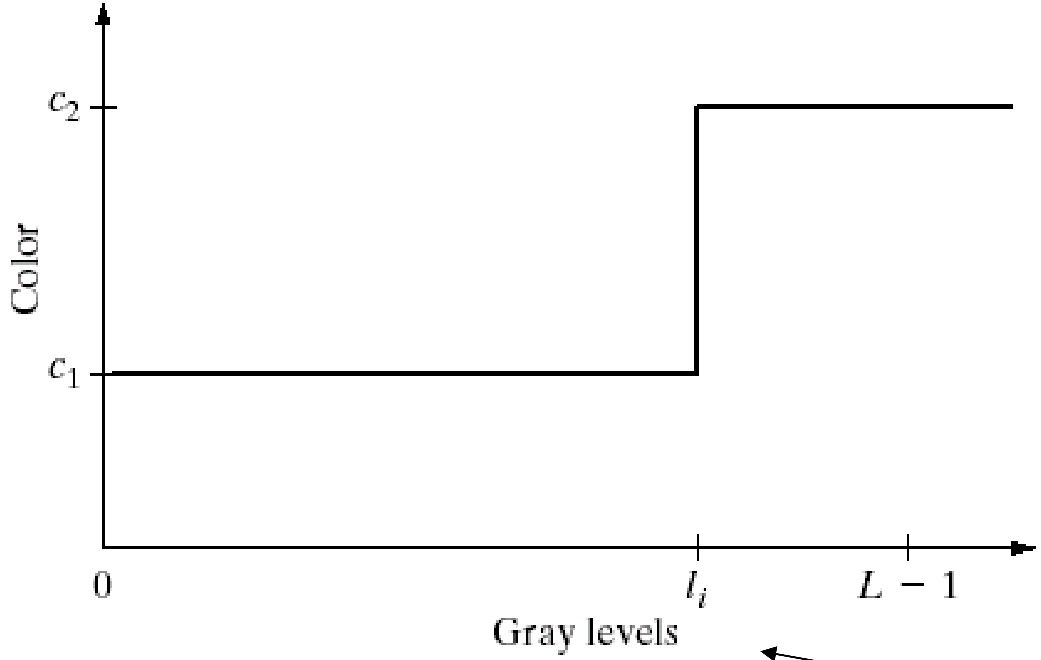


dis-continuous

Un-defined over gray axis

Implementation of colour slicing

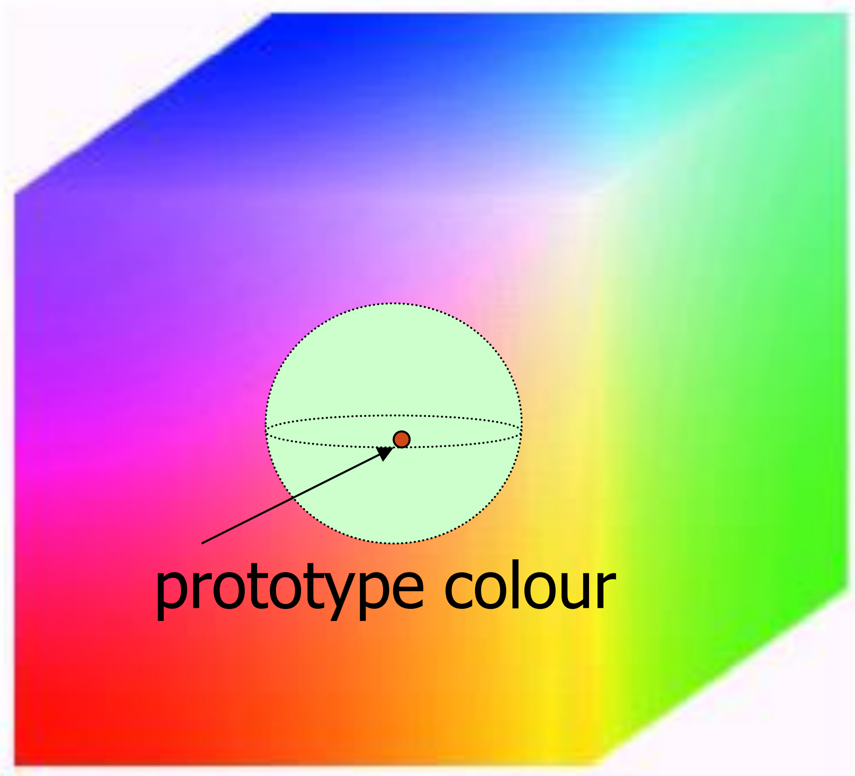
- Recall the pseudo-colour intensity slicing



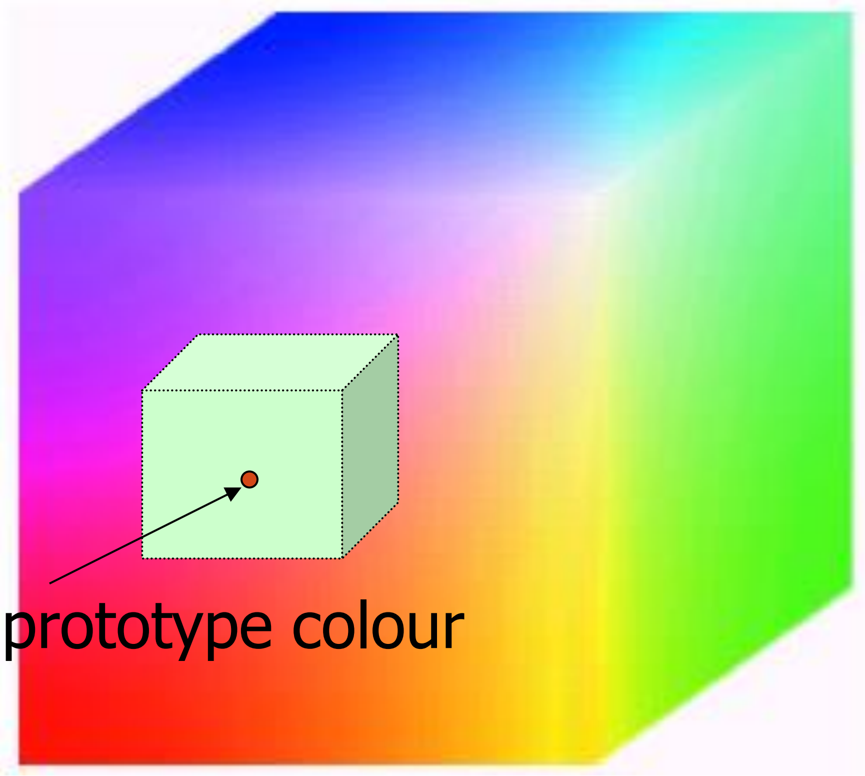
1-D intensity

Implementation of colour slicing

- How to take a region of colours of interest?



Sphere region



Cube region

Application

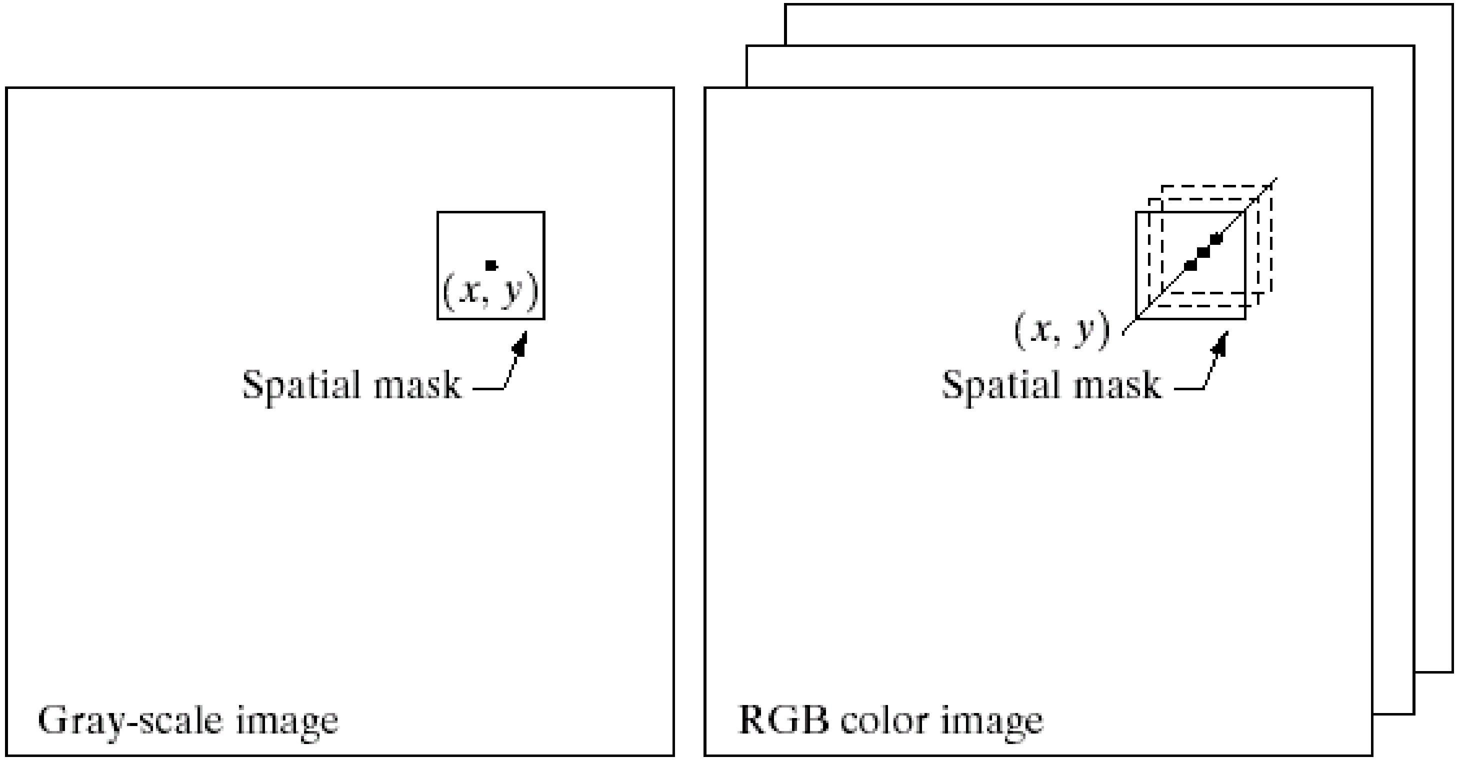
Full color

cube

sphere

colour image smoothing

- Neighborhood processing



colour image smoothing: averaging mask

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

vector processing

Neighborhood
Centered at (x,y)

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

per-component processing


original
R
G
G
H
S
I

Example: 5x5 smoothing mask

RGB model | Smooth I in HSI model | 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.

Pixel Resolution

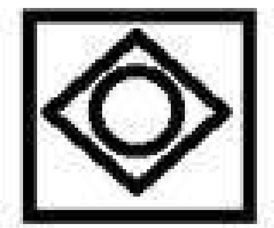
- Resolution
  - The resolution can be defined in many ways. Such as
    - Pixel resolution
    - Spatial resolution,
    - Temporal resolution
    - Spectral resolution
- Megapixels
  - We can calculate mega pixels of a camera using pixel resolution.
  - Column pixels (width) X row pixels (height) / 1 Million.
  - The size of an image can be defined by its pixel resolution.
  - Size = pixel resolution X bpp (bits per pixel)
  - Calculating the mega pixels of the camera
    - Lets say we have an image of dimension: 2500 X 3192.
    - Its pixel resolution = 2500 * 3192 = 7982350 bytes.
    - Dividing it by 1 million = 7.9 = 8 mega pixel (approximately).

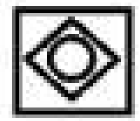
Aspect ratio

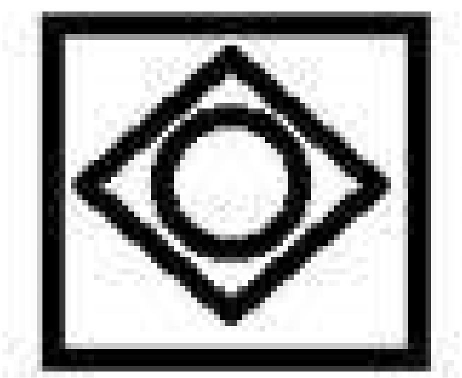
- Another important concept with the pixel resolution is aspect ratio.
  - Aspect ratio is the ratio between width of an image and the height of an image. It is commonly explained as two numbers separated by a colon (8:9).
  - This ratio differs in different images, and in different screens. The common aspect ratios are:
    - 1.33:1, 1.37:1, 1.43:1, 1.50:1, 1.56:1, 1.66:1, 1.75:1, 1.78:1, 1.85:1, 2.00:1, e.t.c
- Advantage
  - Aspect ratio maintains a balance between the appearance of an image on the screen, means it maintains a ratio between horizontal and vertical pixels. It does not let the image to get distorted when aspect ratio is increased.

Example of Aspect Ratio

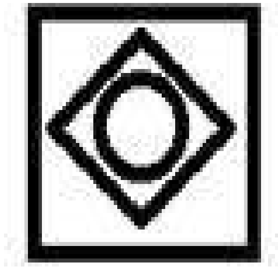
Resized maintaining Aspect Ratio







Resized without maintaining Aspect Ratio



Calculating Dimension of Image

- Given:
  - Aspect ratio: c:r = 6:2
  - Pixel resolution: c * r = 480000
  - Bits per pixel: grayscale image = 8bpp
- Find:
  - Number of rows = ?
  - Number of cols = ?
- Solution

  c = 6r/2....................(1)

  c = 480000/r.............(2)

  From (1) & (2) we can write:

  6r/2= 480000/r

  $\Rightarrow r^2 = (480000*2)/6$

  $\Rightarrow$ r = 400 then c = 1200

  Number of rows = 400

  Number of cols = 1200

Number of rows = 400

Number of cols = 1200

Bits per pixel: grayscale image = 8bpp

Size of the image = rows* cols* bpp

= 400* 1200* 8

= 3840000 bits

- Size of image in bytes = 480000 bytes
- Size of image in kilo bytes = 48 kb *approx*.

Zooming

- Enlarging a picture in a sense that the details in the image became more visible and clear.
- Zooming an image has many wide applications ranging from zooming through a camera lens, to zoom an image on internet e.t.c.
- You can zoom something at two different steps.
  - The first step includes zooming before taking an particular image. This is known as pre processing zoom. This zoom involves hardware and mechanical movement.
  - The second step is to zoom once an image has been captured. It is done through many different algorithms in which we manipulate pixels to zoom in the required portion.
- There are two types of zooming in camera:
  - Optical Zoom
  - Digital Zoom

Camera Zooming

- Optical Zoom
  - The optical zoom is achieved using the movement of the lens of your camera. An optical zoom is actually a true zoom. The result of the optical zoom is far better then that of digital zoom. In optical zoom, an image is magnified by the lens in such a way that the objects in the image appear to be closer to the camera. In optical zoom the lens is physically extend to zoom or magnify an object.
- Digital Zoom:
  - Digital zoom is basically image processing within a camera. During a digital zoom, the center of the image is magnified and the edges of the picture got crop out. Due to magnified center, it looks like that the object is closer to you.
  - During a digital zoom, the pixels got expand , due to which the quality of the image is compromised.
  - The same effect of digital zoom can be seen after the image is taken through your computer by using an image processing toolbox / software, such as Photoshop.

Zooming methods

- Although there are many methods that does this job, but we are going to discuss the most common of them here. They are listed below.
  - Pixel replication or (Nearest neighbor interpolation)
  - Zero order hold method
  - Zooming K times
- Pixel replication
  - It is also known as Nearest neighbor interpolation. As its name suggest, in this method, we just replicate the neighboring pixels. Zooming is nothing but increase amount of sample or pixels. This algorithm works on the same principle.
  - In this method we create new pixels form the already given pixels. Each pixel is replicated in this method n times row wise and column wise and you got a zoomed image. Its as simple as that.

Example of Pixel replication

- If you have an image of 2 rows and 2 columns and you want to zoom it twice or 2 times using pixel replication, here how it can be done. Given image matrix:

1	2
3	4

- Row wise zooming:

1	1	2	2
3	3	4	4

- Column size zooming is done by two steps such as to replicate each of the pixel column wise, that we will simply copy the column pixel to its adjacent new column or simply below it.

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

- New image size: (Original image rows * zooming factor, Original Image cols * zooming factor)

Pixel replication -- Advantage and disadvantage

- One of the advantage of this zooming technique is, it is very simple. You just have to copy the pixels and nothing else.
- The disadvantage of this technique is that image got zoomed but the output is very blurry. And as the zooming factor increased, the image got more and more blurred. That would eventually result in fully blurred image.

Zero order hold Method

- Zero order hold method is another method of zooming. It is also known as zoom twice. Because it can only zoom twice. We will see in the below example that why it does that.
- In zero order hold method, we pick two adjacent elements from the rows respectively and then we add them and divide the result by two, and place their result in between those two elements. We first do this row wise and then we do this column wise.
- New image size
  - As you can see that the dimensions of the new image are 3 x 3 where the original image dimensions are 2 x 2. So it means that the dimensions of the new image are based on the following formula
  - (2(number of rows) minus 1) X (2(number of columns) minus 1)
- Advantages and disadvantage.
  - One of the advantage of this zooming technique , that it does not create as blurry picture as compare to the nearest neighbor interpolation method. But it also has a disadvantage that it can only run on the power of 2. It can be demonstrated here.

Zero order hold Method Example

- Lets take an image of the dimensions of 2 rows and 2 columns and zoom it twice using zero order hold.

1	2
3	4

- Row wise zooming

1	1	2
3	3	4

- As we take the first two numbers : (2 + 1) = 3 and then we divide it by 2, we get 1.5 which is approximated to 1. The same method is applied in the row 2.
- Column wise zooming

1	1	2
2	2	3
3	3	4

- We take two adjacent column pixel values which are 1 and 3. We add them and got 4. 4 is then divided by 2 and we get 2 which is placed in between them. The same method is applied in all the columns.

K-Times zooming Method

- K times is the third zooming method we are going to discuss. It is one of the most perfect zooming algorithm discussed so far. It caters the challenges of both twice zooming and pixel replication. K in this zooming algorithm stands for zooming factor.
- It works like this way.
  - First of all, you have to take two adjacent pixels as you did in the zooming twice. Then you have to subtract the smaller from the greater one. We call this output (OP).
  - Divide the output(OP) with the zooming factor(K). Now you have to add the result to the smaller value and put the result in between those two values.
  - Add the value OP again to the value you just put and place it again next to the previous putted value. You have to do it till you place k-1 values in it.
  - Repeat the same step for all the rows and the columns , and you get a zoomed images.
- New image size =(K (number of rows minus 1) + 1) X (K (number of cols minus 1) + 1)
- Advantages and disadvantages
  - The one of the clear advantage that k time zooming algorithm has that it is able to compute zoom of any factor which was the power of pixel replication algorithm , also it gives improved result (less blurry) which was the power of zero order hold method. So hence It comprises the power of the two algorithms.
  - The only difficulty this algorithm has that it has to be sort in the end, which is an additional step, and thus increases the cost of computation.

Example

15	30	15
30	15	30

Table 1: Row –wise Zooming

15	20	25	30	20	25	15
30	20	25	15	20	25	30

Table 2: Row –wise Zooming

15	20	25	30	25	20	15
30	25	20	15	20	25	30

Column-wise Zooming

15	20	25	30	25	20	15
20	21	21	25	21	21	20
25	22	22	20	22	22	25
30	25	20	15	20	25	30

the dimensions of the new image are 4 x 7.

Thank You!