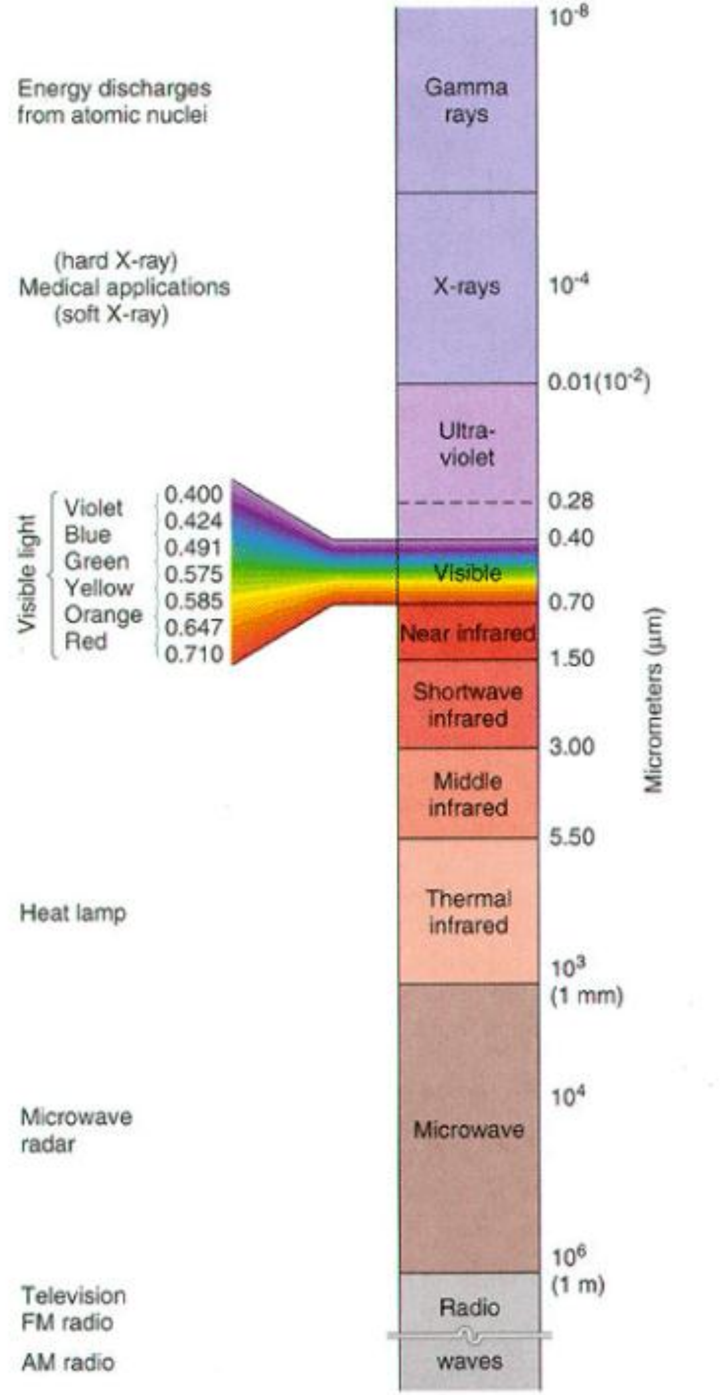
Colour Image Processing

- Colour plays an important role in image processing
- Colour image processing can be divided into two major areas
  - Full-colour processing: Colour sensors such as colour cameras and colour scanners are used to capture coloured image. Processing involves enhancement and other image processing tasks
  - Pseudo-colour processing : Assigning a colour to a particular monochrome intensity range of intensities to enhance visual discrimination.

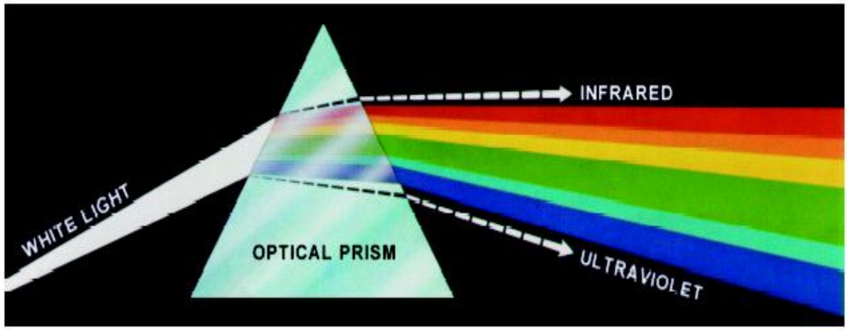
Physical Background

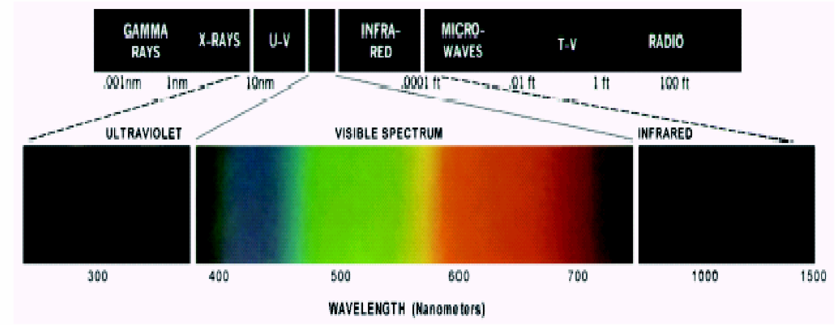
- Visible light: a narrow band of electromagnetic radiation → **380nm (blue)** – **780nm (red)**
- Wavelength: Each physically distinct colour corresponds to at least one wavelength in this band.
- Pure Colours: Pure or monochromatic colours do not exist in nature.



source: Christopherson (2000) *Geosystems*

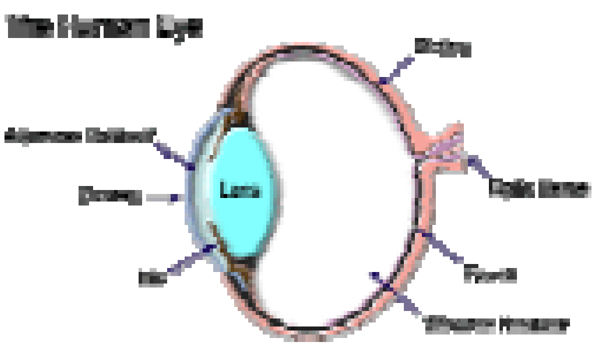
- Visible spectrum: approx. 400 ~ 700 nm
- The frequency or mix of frequencies of the light determines the colour
- Visible colours: VIBGYOR with UV and IR at the two extremes (excluding)

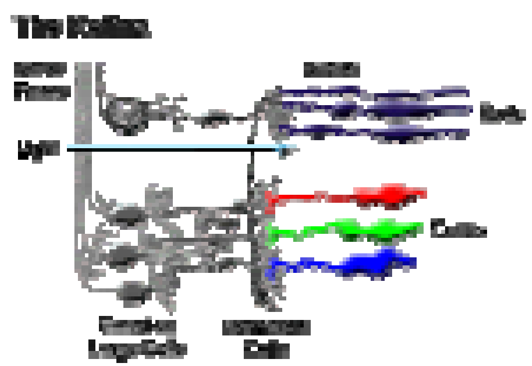




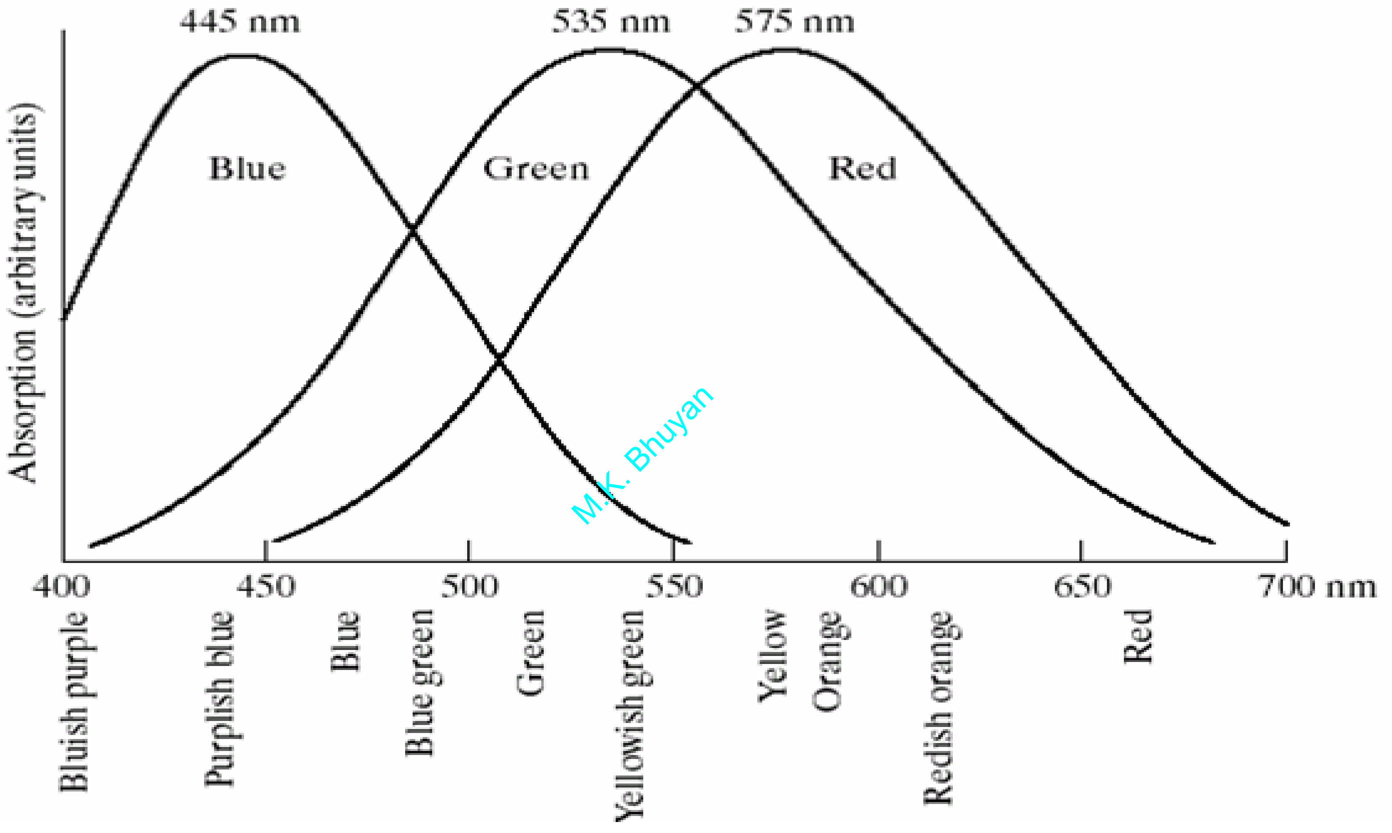
Human Vision

- Human retina have two types of sensors:
  - Cone
    - Sensitive to *colored light*, but not very sensitive to *dim light*
    - Three types of cones (correspond to R, G, B)
  - Rod
    - (Very) sensitive to *achromatic light*
    - Important for vision, play no role in image reproduction





- Cones are the sensors in the eye responsible for colour vision
- Humans perceive colour using three types of cones
- **Primary colours: RGB** – because the cones of our eyes can basically absorb these three colours.
- The sensation of a certain colour is produced due to the mixed response of these three types of cones in a certain proportion
- Experiments show that 6-7 million cones in the human eye can be divided into red, green and blue vision.
- 65% cones are sensitive to red vision, 33% are for green and only 2% are for blue vision (blue cones are the most sensitive)



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

- According to the CIE (Commission Internationale de l'Eclairage, The International Commission on Illumination) the wavelength of each primary colour is set as follows:
  blue=435.8nm, green=546.1nm, and red=700nm.

- However this standard is just an approximate; it has been found experimentally that no single colour may be called red, green, or blue There is no pure red, green or blue colour.

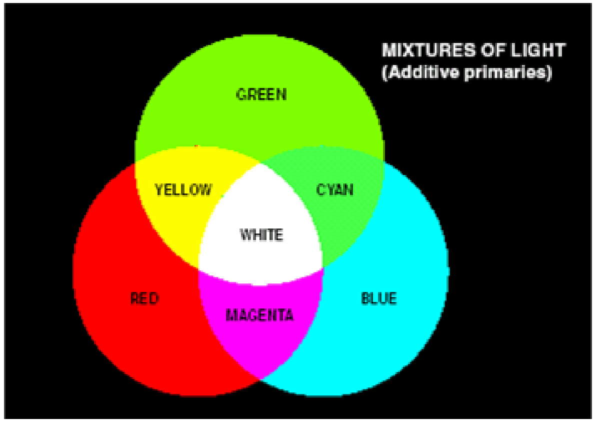
- The primary colours can be added in certain proportions to produce different colours of light.

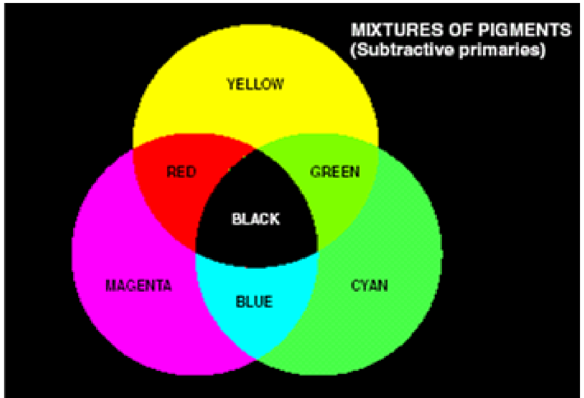
- The colour produced by mixing RGB is not a natural colour.
- A natural colour will have a single wavelength, say λ.
- On the other hand, the same colour is artificially produced by combining weighted R, G and B each having different wavelength.
- The idea is that these three colours together will produce the same amount of response as that would have been produced by wavelength λ alone (proportion of RGB is taken accordingly), thereby giving the sensation of the colour with wavelength λ to some extent.

- Mixing two primary colours in equal proportion produces a secondary colour of light: magenta (R+B), cyan (G+B) and yellow (R+G).

- Mixing RGB in equal proportion produces white light.

- The second figure shows primary/secondary colours of pigments.





PRIMARY AND SECONDARY COLORS
OF LIGHT AND PIGMENT

- Brightness perceived (subjective brightness) is a logarithmic function of light intensity. In other words it embodies the chromatic notion of intensity.

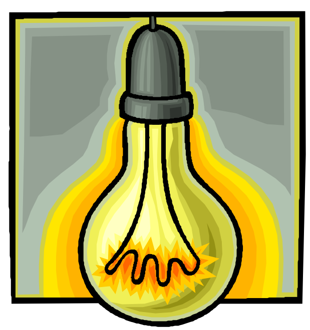
- Hue is an attribute associated with the dominant wavelength in a mixture of light waves. It represents the dominant colour as perceived by an observer. Thus, when we call an object red, orange, or yellow, we are specifying its hue.

- Saturation refers to the relative purity or the amount of white light mixed with hue. The pure spectrum colours are fully saturated. colour such as pink (red and white) is less saturated. The degree of saturation is inversely proportional to the amount of white light added.

- Red, Green, Blue, Yellow, Orange, etc. are different hues. Red and Pink have the same hue, but different saturation. A faint red and a piercing intense red have different brightness.
- Hue and saturation taken together are called chromaticity.
- So, brightness + chromaticity defines any colour.

Problems with Processing Colour Images

- When processing colour images, the following problems (amongst others) have to be dealt with:
  - The images are vectorial → 3 numbers are associated with each pixel.
  - The colours recorded by a camera are heavily dependent on the lighting conditions.

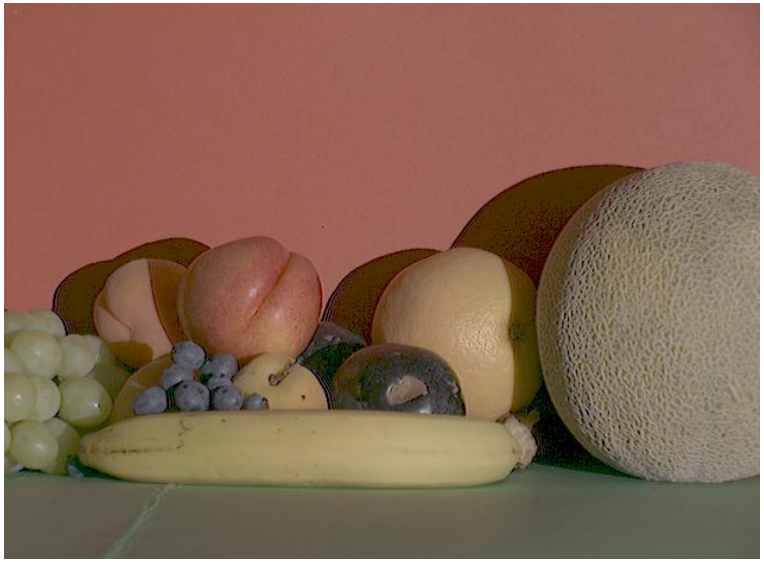
Lighting conditions

- The lighting conditions of the scene have a large effect on the colours recorded.

Image taken lit by a flash.

Image taken lit by a tungsten lamp.

The following four images of the same scene were acquired under different lighting conditions


Dealing with Lighting Changes

- Knowing just the RGB values is not enough to know everything about the image.
  - The R, G and B primaries used by different devices are usually different.
- For scientific work, the camera and lighting should be calibrated.
- For multimedia applications, this is more difficult to organise:
  - Algorithms exist for estimating the illumination colour.

XYZ Colour System

- CIE (Commision Internationale de L'Eclairage), 1931. Spectral RGB primaries (scaled, such that *X*=*Y*=*Z* matches spectrally flat white).
- The entire colour gamut can be produced by the three primaries used in CIE 3-colour system. A particular colour (of wavelength λ) be represented by three components *X*, *Y*, and *Z*. These are called tri-stimulus values.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.210 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R_\lambda \\ G_\lambda \\ B_\lambda \end{bmatrix}$$

λ denotes corresponding spectral component

XYZ Color System

- A colour is then specified by its tri-chromatic coefficients, defined as

  $x = X/(X+Y+Z)$ $y = Y/(X+Y+Z)$

  $z = Z/(X+Y+Z)$

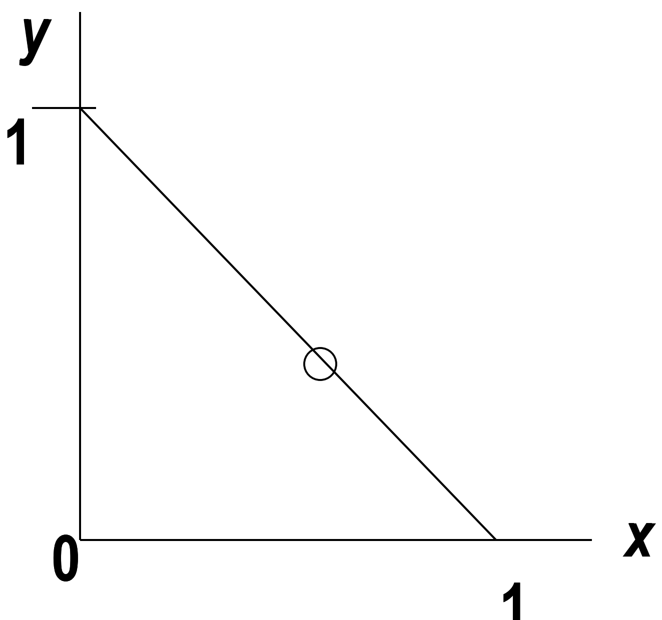
  so that

  $x + y + z = 1$

- For any wavelength of light in the visible spectrum, these values can be obtained directly from curves or tables compiled from experimental results.

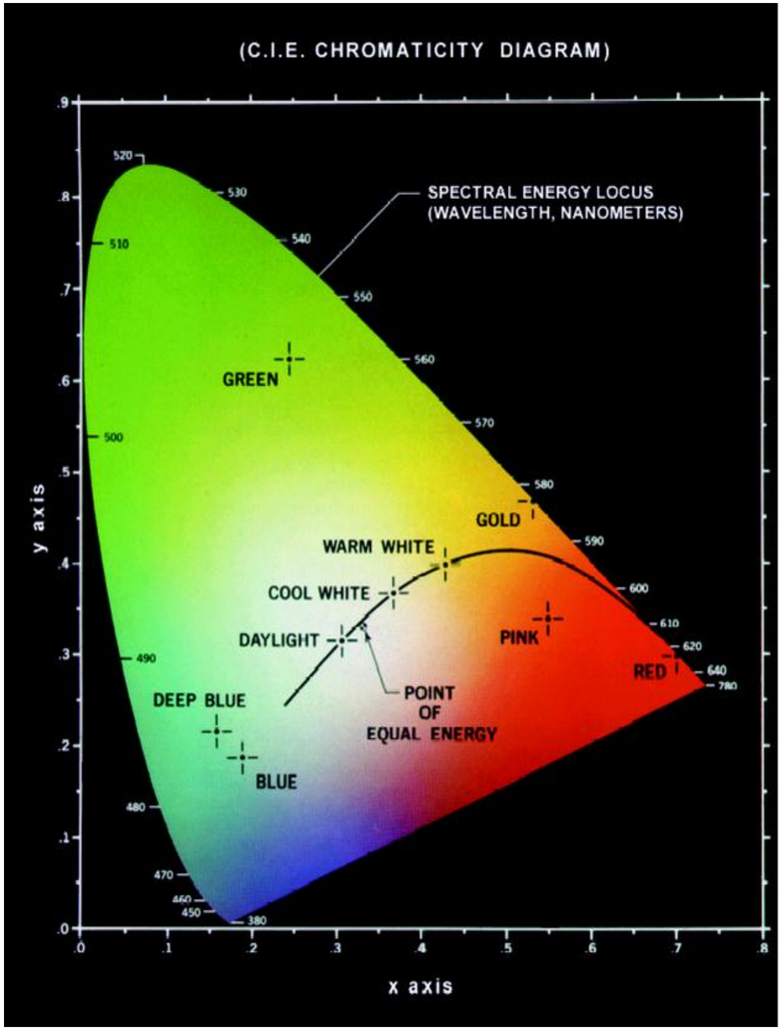
Chromaticity Diagram

- Shows colour composition as a function of *x* and *y* (only two of *x*, *y* and z are independent → $z = 1 - (x + y)$ and so not independent of them)
- The triangle in the diagram below shows the colour gamut for a typical RGB system plotted as the XYZ system.



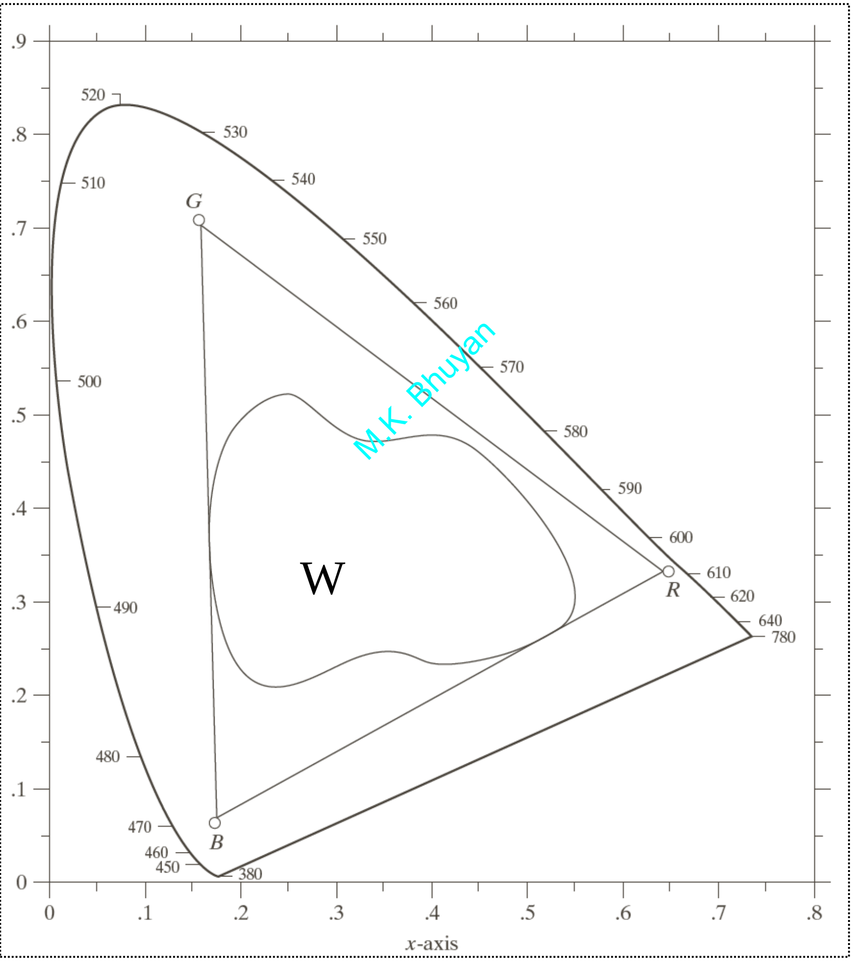
The axes extend from 0 to 1. The origin corresponds to BLUE. The extreme points on the axes correspond to RED and GREEN.The point corresponding to *x*= y= 1/3 (marked by the white spot) corresponds to WHITE.

- The positions of various spectrum colours from violet (380nm) to red (700 nm) are indicated around the boundary (100% saturation). These are *pure* colours.
- Any inside point represents mixture of spectrum colours.
- A straight line joining a spectrum colour point to the equal energy point shows all the different shades of the spectrum colour.



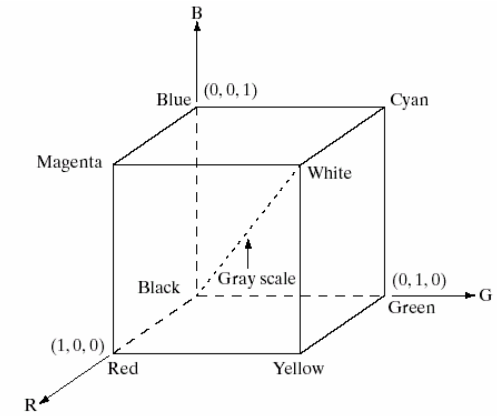
- Any color in the interior of the “horse shoe" can be achieved through the linear combination of two pure spectral colors
- A straight line joining any two points shows all the different colours that may be produced by mixing the two colours corresponding to the two points
- The straight line connecting red and blue is referred to as the *line of purples*

RGB primaries form a triangular color gamut. The white colour falls in the center of the diagram

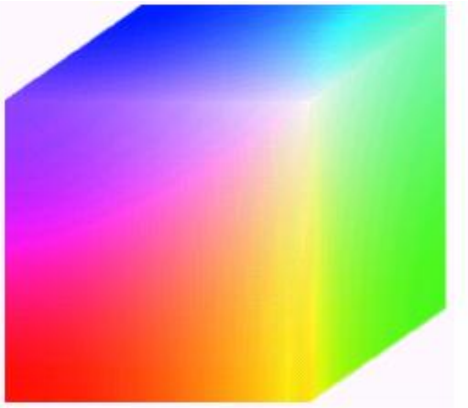


RGB Colour Model

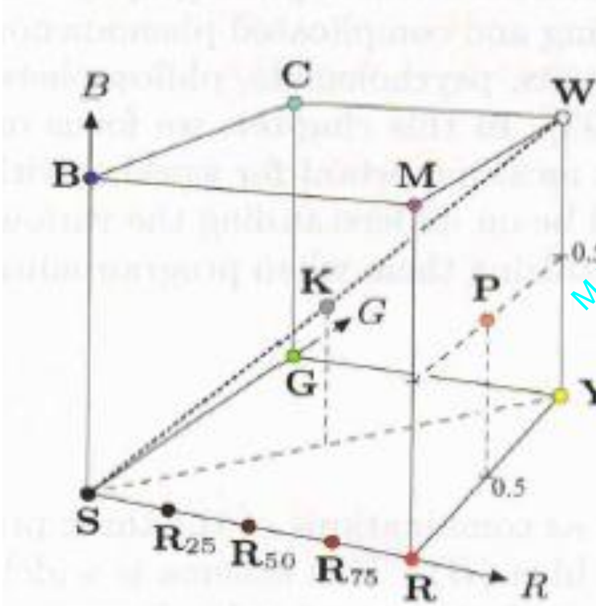
- Colour models are normally invented for practical reasons, and so a wide variety exist.
- The RGB colour space (model) is a linear colour space that formally uses single wavelength primaries.
- Informally, RGB uses whatever phosphors a monitor has as primaries
- Available colours are usually represented as a unit cube — usually called the RGB cube — whose edges represent the R, G, and B weights.



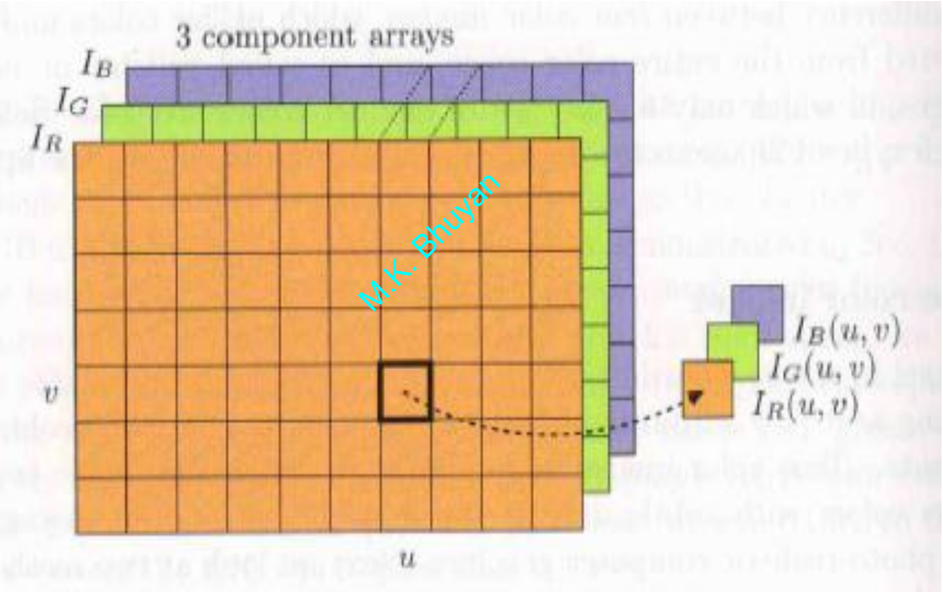
Schematic of the RGB colour cube

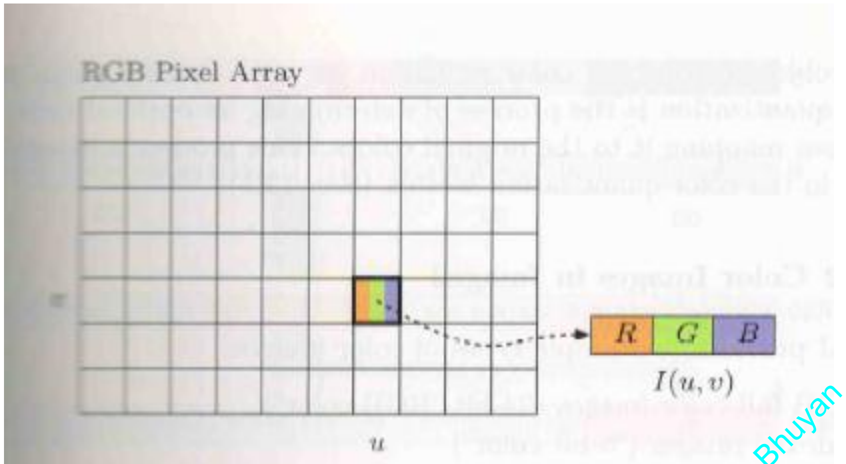


RGB 24-bit colour cube



Point	Color	RGB Value R	G	B
S	Black	0.00	0.00	0.00
R	Red	1.00	0.00	0.00
Y	Yellow	1.00	1.00	0.00
G	Green	0.00	1.00	0.00
C	Cyan	0.00	1.00	1.00
B	Blue	0.00	0.00	1.00
M	Magenta	1.00	0.00	1.00
W	White	1.00	1.00	1.00
K	50% Gray	0.50	0.50	0.50
R_{75}	75% Red	0.75	0.00	0.00
R_{50}	50% Red	0.50	0.00	0.00
R_{25}	25% Red	0.25	0.00	0.00
P	Pink	1.00	0.50	0.50


3 component arrays
I_B
I_G
I_R
M.K. Bhuyan
v
u
$I_B(u, v)$
$I_G(u, v)$
$I_R(u, v)$


RGB Pixel Array
R
G
B
I(u, v)
u
v

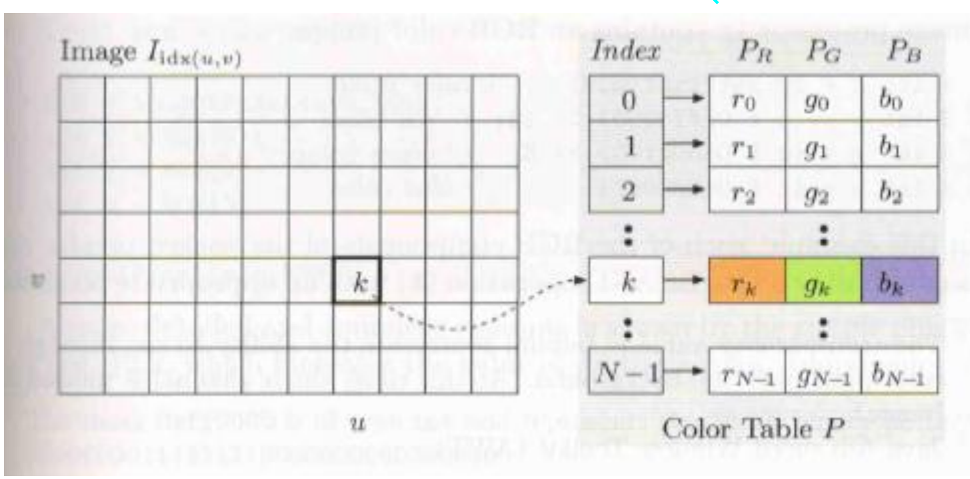


Image I_idx(u,v)
Index
P_R
P_G
P_B
0
1
2
k
N−1
r_0 g_0 b_0
r_1 g_1 b_1
r_2 g_2 b_2
r_k g_k b_k
r_N−1 g_N−1 b_N−1
u
v
Color Table P

CMY and CMYK colour models

Cyan, Magenta, Yellow Primary pigment colour

Subtractive color space

Related to RGB by

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

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Should produce black

Practical printing devices additional black pigment is needed. This gives the CMYK colour space

Decoupling the colour components from intensity

Decoupling the intensity from colour components has several advantages:

- Human eyes are more sensitive to the intensity than to the hue
- We can distribute the bits for encoding in a more effective way.
- We can drop the colour part altogether if we want gray-scale images.
- In this way, black-and-white TVs can pick up the same signal as color ones.
- We can do image processing on the intensity and color parts separately.

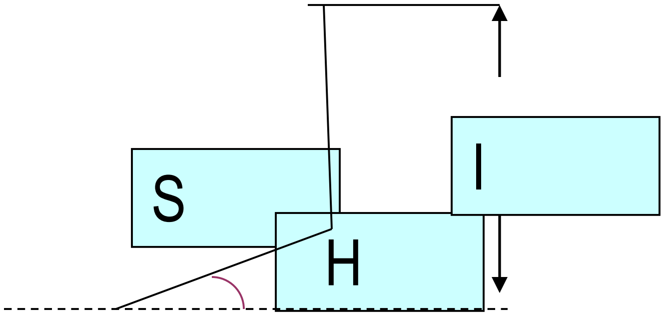
Example:

Histogram equalization on the intensity part to contrast enhance the image while leaving the relative colors the same

HSI Colour system

- Hue is the colour corresponding to the dominant wavelength measured in angle with reference to the red axis
- Saturation measures the purity of the colour. In this sense impurity means how much white is present.

  Saturation is 1 for a pure colour and less than 1 for an impure colour.
- Intensity is the chromatic equivalent of brightness also means the grey level component.



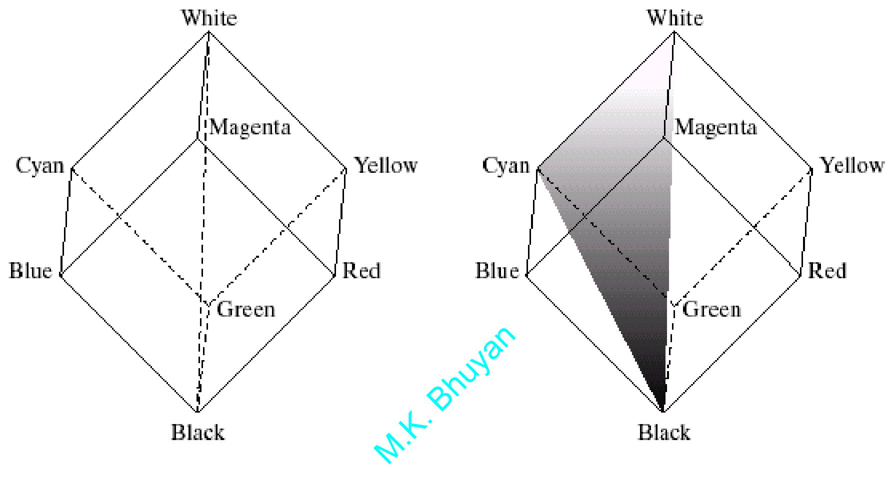
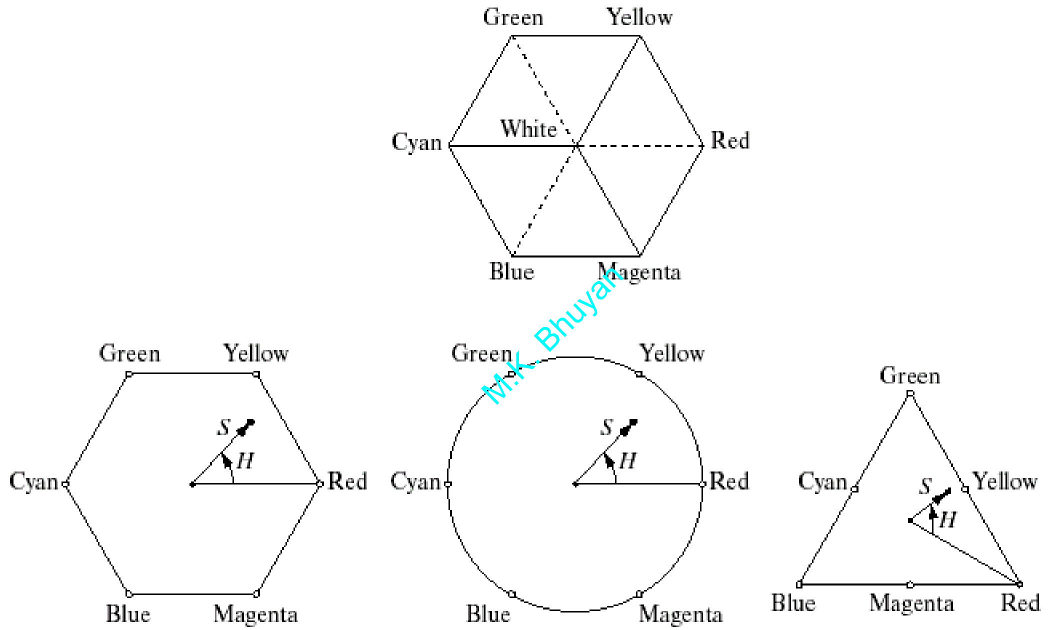


FIGURE 6.12 Conceptual relationships between the RGB and HSI color models.

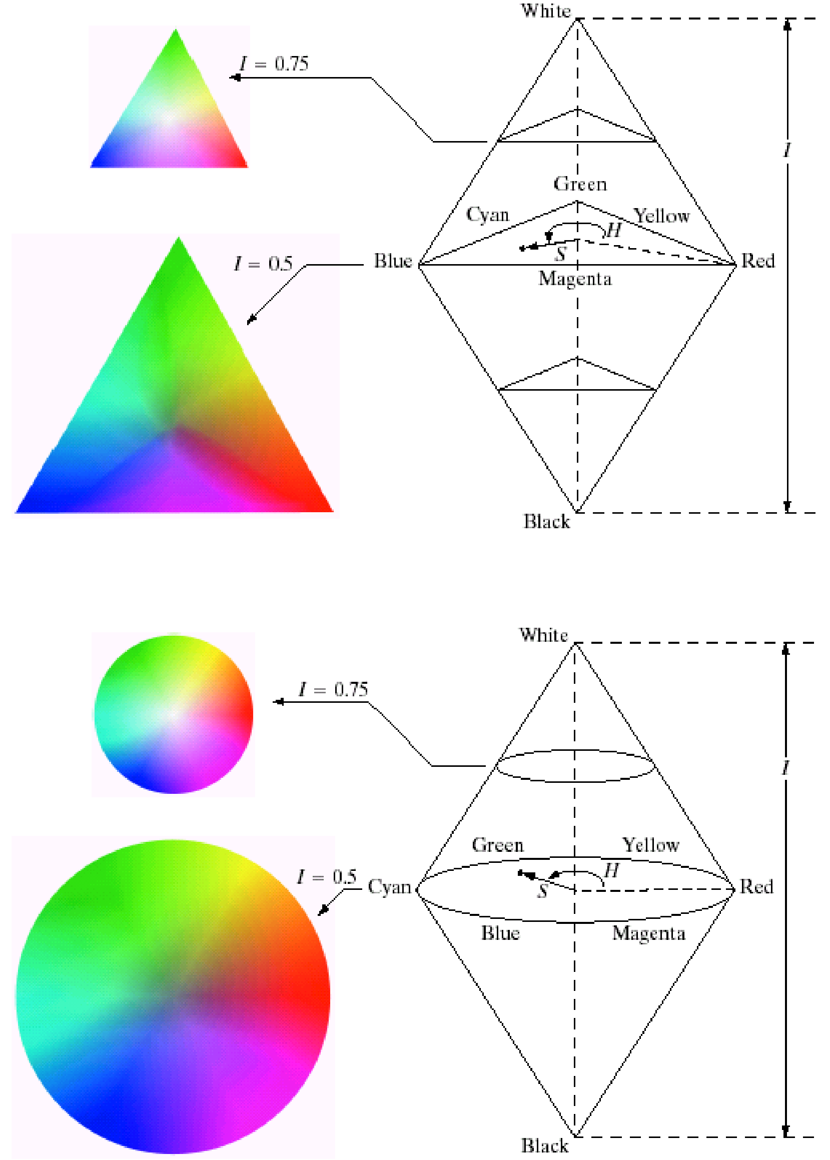
- HSI model can be obtained from the RGB model.
- The diagonal of the joining Black and White in the RGB cube is the intensity axis.
- All points contained in the plane segment defined by the intensity axis and the boundaries of the cube have the same hue.
- Saturation (purity) of a color increases as a function of distance from the intensity axis. Saturation of points on the intensity axis is zero.



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.

- HSI colour model based on a triangle and a circle are shown
- The circle and the triangle are perpendicular to the intensity axis.



The following formulae show how to convert from RGB space to HSI:

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360^o - \theta & \text{if } B > G \end{cases}$$

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

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

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

To convert from HSI to RGB, the process depends on which colour sector H lies in.

For the RG sector: $0^o \le H \le 120^o$

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

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

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

For the *GB* sector: $120^o \le H \le 240^o$

$$H = H - 120^\circ$$

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

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

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

For the *BR* sector: $240^o \le H \le 360^o$

$$H = H - 240^\circ$$

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

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

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

YIQ model

- YIQ colour model is the NTSC standard for analog video transmission.

    Y stands for intensity (provide all video information required for the monochrome TV).

    I is the in phase component, orange-cyan axis

    Q is the quadrature component, magenta-green axis

- Y component is decoupled because the signal has to be made compatible for both monochrome and colour television.
- The relationship between the YIQ and RGB model is

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.581 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YIQ model is designed to take the advantage of human visual system's greater sensitivity to changes in luminance than to the changes in hue or saturation. Luminance is proportional to the amount of light perceived by the eye. So, importance of YIQ model is that luminance component of an image can be processed without affecting its color content.

Y-Cb-Cr colour model

- International standard for studio-quality video
- Y is the intensity corresponding to YIQ model.
- C_b and C_r are so selected that the resulting scheme is efficient for compression (Less spectral redundancy between C_b & C_r i.e., co-efficient are less correlated.
- This colour model is chosen in such a way that it achieves maximum amount of decorrelation.
- This colour model is obtained by extensive experiments on human observers

$$Y = 0.299R + 0.587G + 0.114B$$

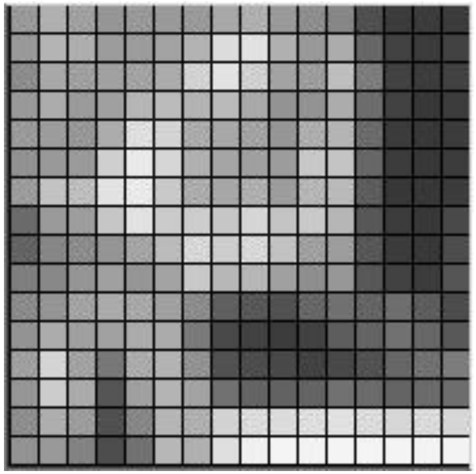
$$C_b = B - Y$$

$$C_r = R - Y$$

Processing Vectorial Images

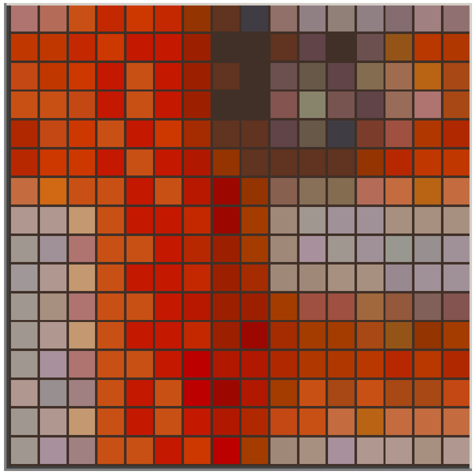
- A vectorial image has a vector at each pixel. For colour images, these vectors each have 3 components.
- Vectorial images with larger numbers of components also exist, e.g. in satellite imagery.
- There are two ways one can process vectorial images:
  - Marginal processing.
  - Vectorial processing.

Greyscale



$f(x, y) \rightarrow \{0, 1, \ldots, N\}$

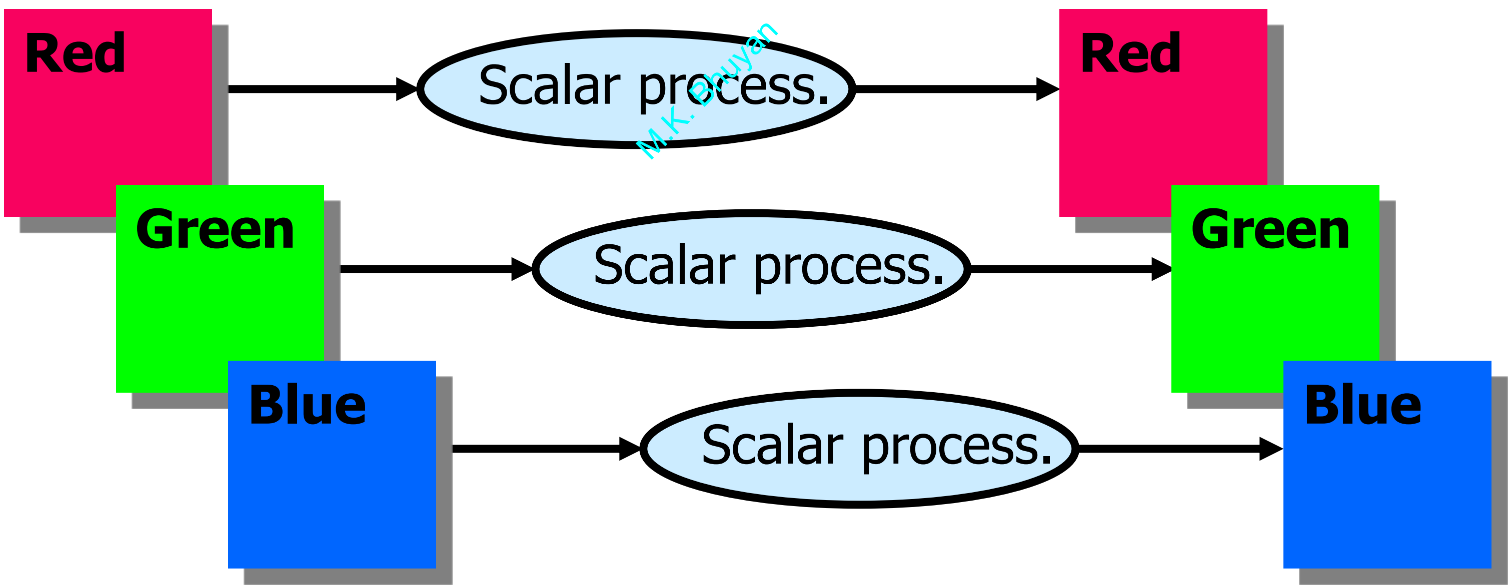
Colour



$f(x, y) \rightarrow [\ \{0, \ldots, N\}, \{0, \ldots, N\}, \{0, \ldots, N\}\ ]$

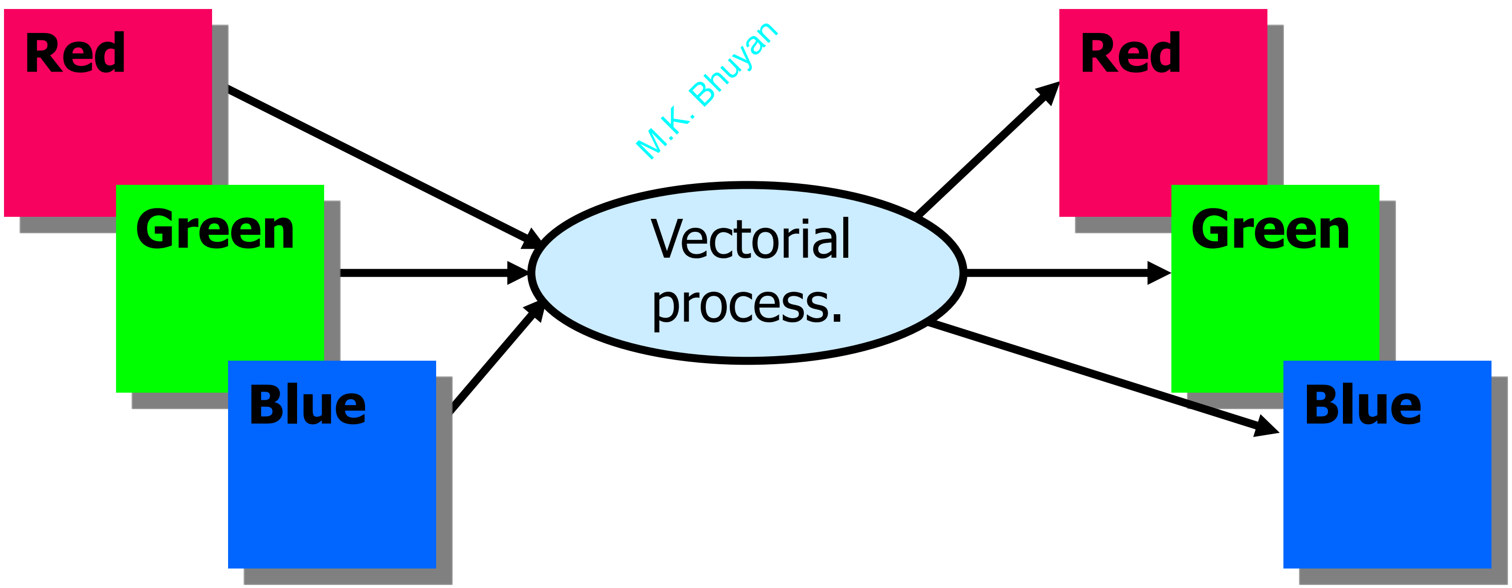
Marginal Processing

- Each channel is processed separately:

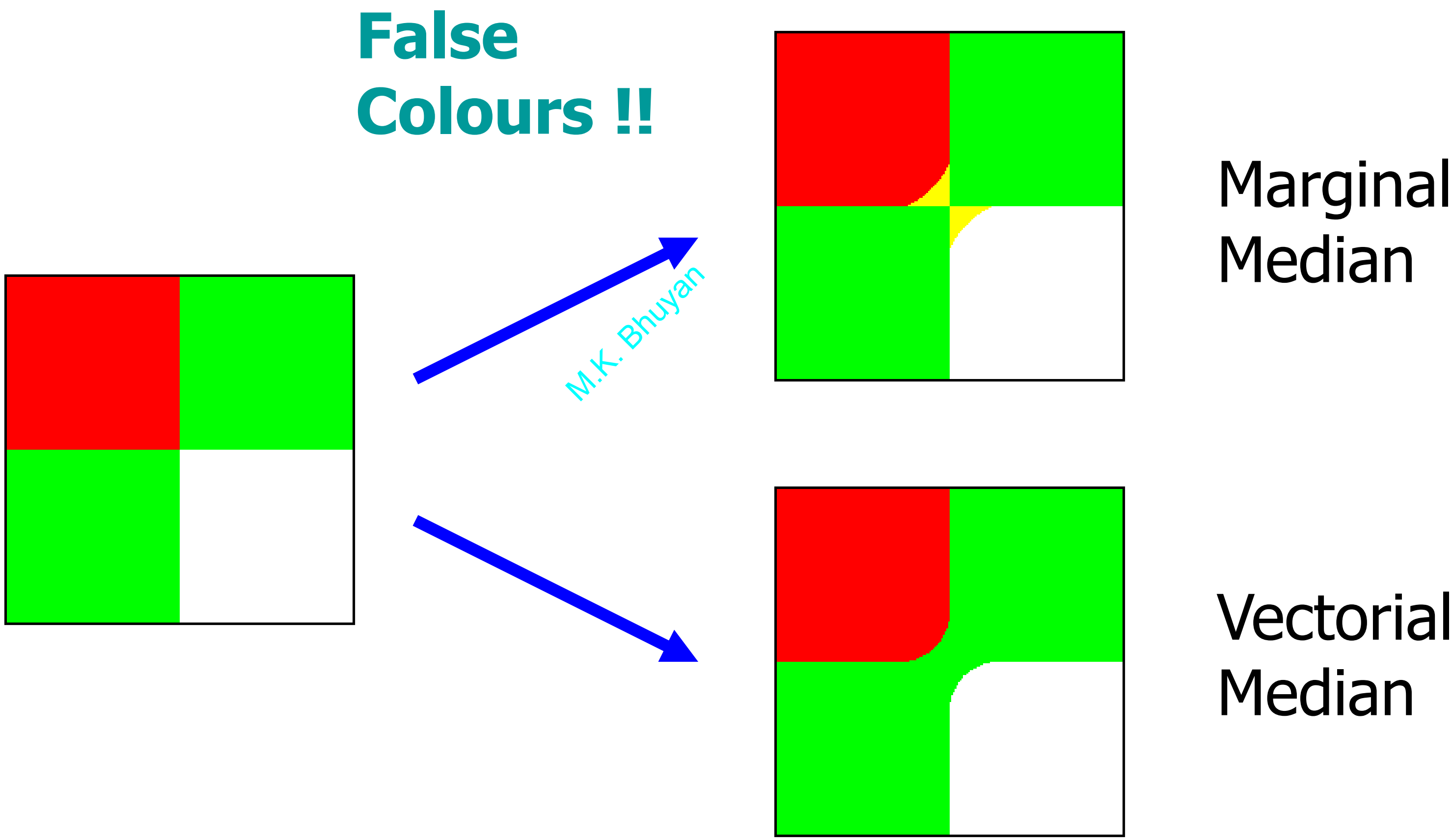


Vectorial Processing

- The colour triplets are processed as single units:



The Problem of False Colours



The Problem of False Colours

$$\begin{pmatrix} R=100 \\ G=0 \\ B=0 \end{pmatrix} \quad \begin{pmatrix} R=0 \\ G=100 \\ B=0 \end{pmatrix} \quad \begin{pmatrix} R=100 \\ G=100 \\ B=100 \end{pmatrix}$$

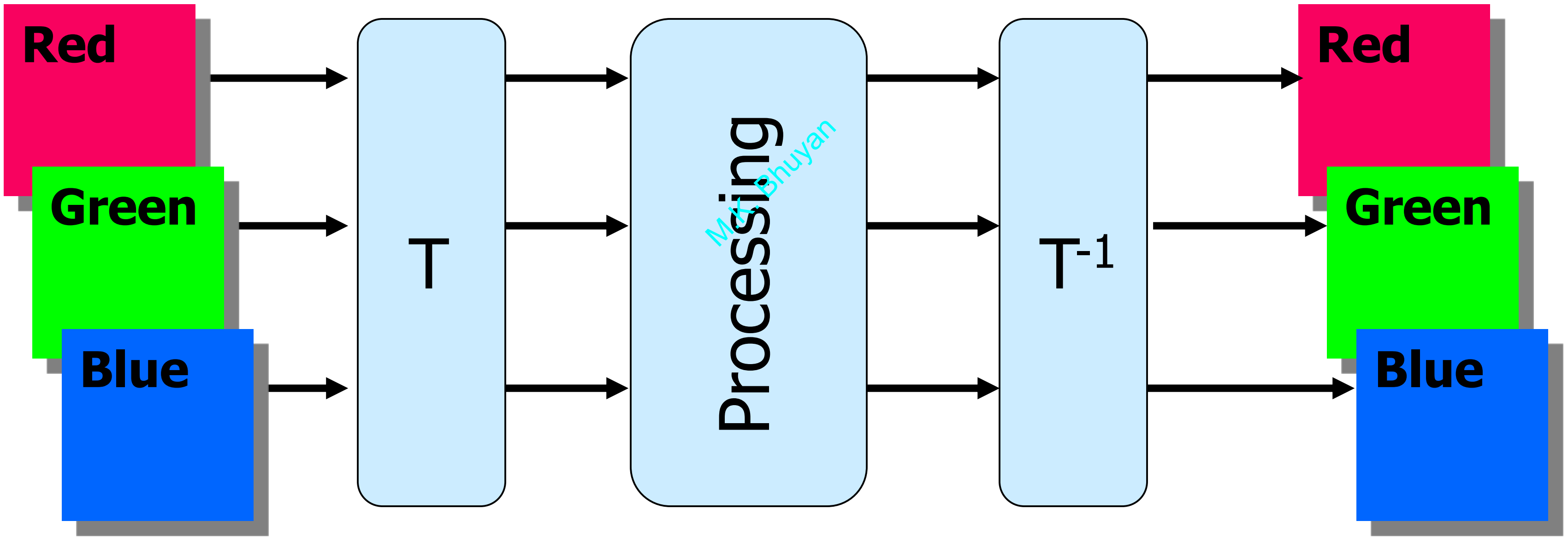
Marginal median

$$\begin{pmatrix} R=100 \\ G=100 \\ B=0 \end{pmatrix}$$

Vectorial median

$$\begin{pmatrix} R=0 \\ G=0 \\ B=100 \end{pmatrix}$$

Processing Strategy



Processing of Color Images

- Two methods to use
  - Process each R, G, B matrix separately
  - Transform the color space to one in which the intensity is separated from the color, and process the intensity component only

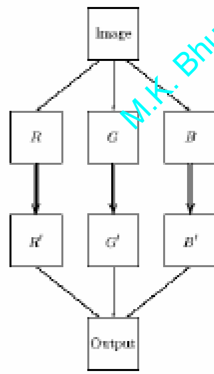


Figure 11.15: RGB processing

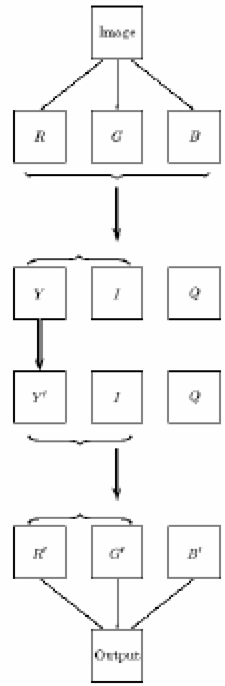


Figure 11.16: Intensity processing

Contrast Enhancement

```
c = imread('cat.tif');
imshow(c)

cn = rgb2ntsc(c);

% processing the intensity component only!
cn(:,:,1) = histeq(cn(:,:,1));
c2 = ntsc2rgb(cn);
figure, imshow(c2)

% processing each of the RGB components
cr = histeq(c(:,:,1));
cg = histeq(c(:,:,2));
cb = histeq(c(:,:,3));
c3 = cat(3,cr,cg,cb);
figure, imshow(c3)

% Compare which one is better?!
```

Spatial Filtering

```
c = imread('cat.tif');
imshow(c)

% filtering on each of the RGB components
a15 = fspecial('average',15);
cr = imfilter(c(:,:,1),a15);
cg = imfilter(c(:,:,2),a15);
cb = imfilter(c(:,:,3),a15);
blur = cat(3,cr,cg,cb);
figure, imshow(blur)

% filtering on the intensity component only
cn = rgb2ntsc(c);
a = fspecial('unsharp');
cn(:,:,1) = filter2(a,cn(:,:,1));
cu = ntsc2rgb(cn);
figure,imshow(cu);

% Which one is better?!
```


Noise Reduction

```
tw = imread('twins.tif');

tn = imnoise(tw,'salt & pepper');
Figure, imshow(tn);
figure, imshow(tn(:,:,1))
figure, imshow(tn(:,:,2))
figure, imshow(tn(:,:,3))

% filtering on each of the RGB components
trm = medfilt2(tn(:,:,1));
tgm = medfilt2(tn(:,:,2));
tbm = medfilt2(tn(:,:,3));
tm = cat(3,trm,tgm,tbm);
figure,imshow(tm)

% filtering on intensity component only
tnn = rgb2ntsc(tn);
tnn(:,:,1) = medfilt2(tnn(:,:,1));
tm2 = ntsc2rgb(tnn);
figure,imshow(tm2)

% Which one is better?!
```


Edge Detection

```
f = imread('flowers.tif'); imshow(f)

% Edge detection on intensity component only
fg = rgb2gray(f);
fe1 = edge(fg);
figure,imshow(fe1)

% Edge detection on each of the RGB components
f1 = edge(f(:,:,1));
f2 = edge(f(:,:,2));
f3 = edge(f(:,:,3));
fe2 = f1 | f2 | f3;
figure,imshow(fe2)
```



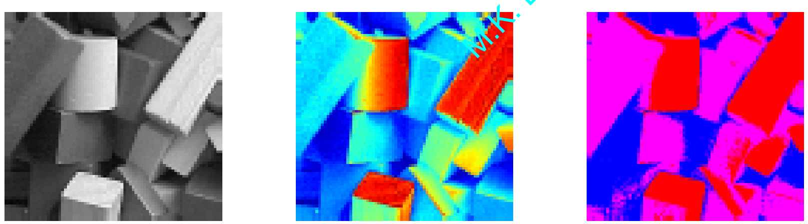

Pseudo Colour Image Processing

❖ To improve the visual quality of the image. Pseudo colouring scheme converts the gray level image into RGB image (not unique).

❖ This method produces a composite image whose colour content is modulated by the nature of the transformation functions. These are transformations on the gray-level values of an image and are not functions of position.

Pseudocoloring

- Intensity slicing (for medical images)
  - Break up the image into various grey level ranges and assign a different color to each range
- Gray to color transformation
  - Three functions $f_R(x)$, $f_G(x)$, $f_B(x)$ assigning red, green and blue values to each grey level and used for display



Colour Balancing

Refers to the adjustment of the relative amounts of red, green, and blue primary colors in an image such that neutral colors are reproduced correctly. Colour imbalance is a serious problem in Colour Image Processing

- Select a gray level, say white, where RGB components are equal
- Examine the RGB values. Find the transformation to make R=G=B. Keep one component fixed and match the other components to it, there by defining a transformation for each of the variable components
- Apply the transformation to balance the entire image.

Example: Colour Balanced Image

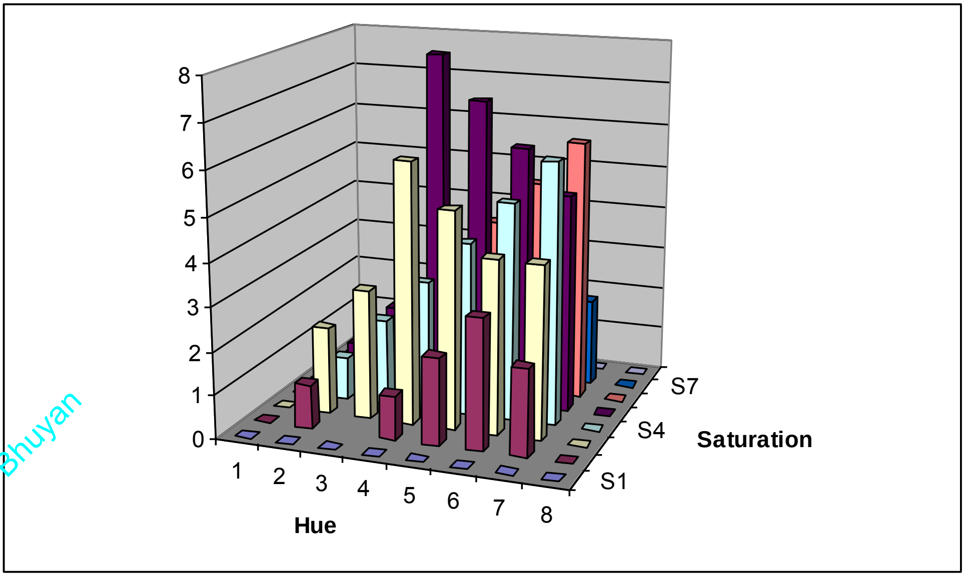
Histograms of a colour image

-Histogram of Luminance and chrominance components separately

-Colour histograms (H-S components or normalized R-G components

-Useful way to segment objects like skin non-skin

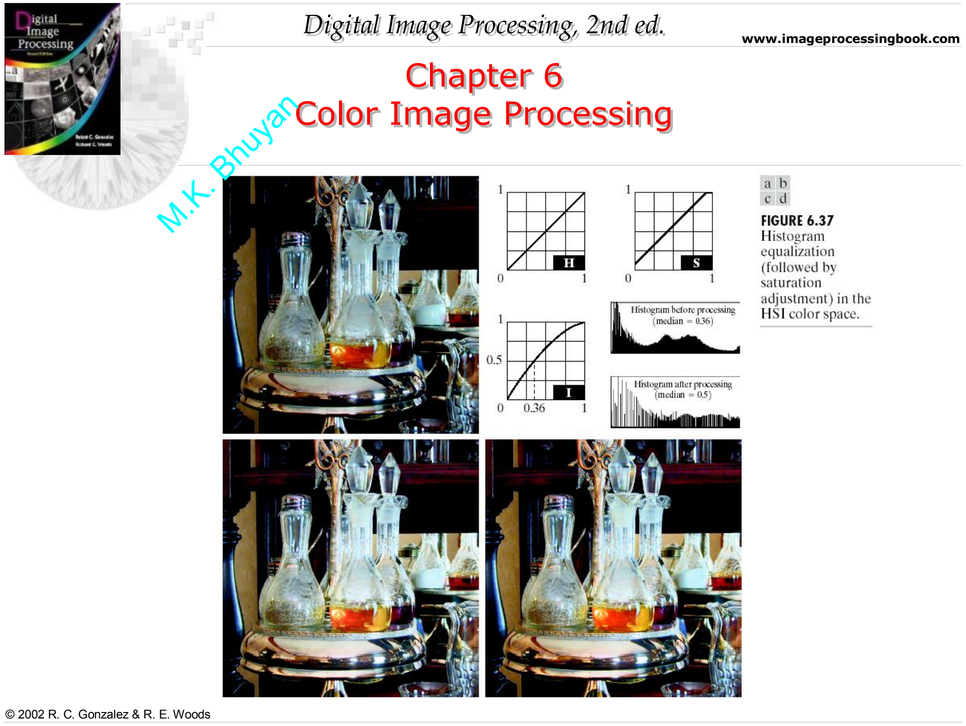
-Colour based indexing of images



Contrast enhancement by histogram equalisation

Histogram equalisation cannot be applied separately for each channel

- Convert to HIS space
- Apply histogram equalisation to the I component
- Correct the saturation if needed
- Convert back to RGB values



Colour image smoothing

- Vector processing is used
- Averaging in vector is equivalent to averaging separately in each channel

Example- Averaging low pass filter: Averaging a vector is equivalent to averaging all the components

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.

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.

MEDIAN FILTER ON COLOR IMAGES

•We cannot apply the median filtering to the component images separately because that will result in colour distortion

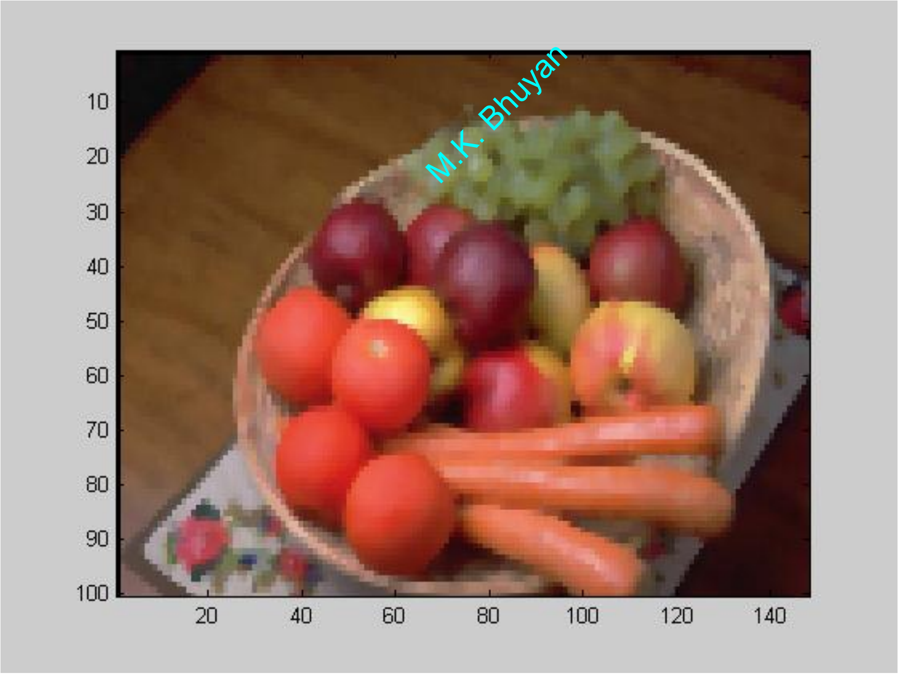
•If each channel is separately median filtered then the net median will be completely different from the values of the pixel in the window

MEDIAN FILTER ON COLOR IMAGES

The original image

MEDIAN FILTER ON COLOR IMAGES

The resulting image after applying the median filter separately on each matrix



MEDIAN FILTER ON COLOR IMAGES

- **The second application of median filter on color images is the vector median filter.**
- **This time, the filter is applied on each matrix element treated as a vector.**

Vector median filter

•Vector median filter will minimize the sum of the distances of a vector pixel from the other vector pixels in the window

•The pixel with the minimum distance will give the vector median.

•The set of all vector pixels inside the window is given by

$$\boldsymbol{X}_{\mathrm{W}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$$

Computation of vector median filter

(1) Find the sum of the distances δ_i of the ith $(1 \le i \le N)$ vector pixel from all other neighbouring vector pixels in the window given by

$$\delta_i = \sum_{j=1}^{N} d(\mathbf{x}_i, \mathbf{x}_j)$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ represents an appropriate distance measure between the ith and jth neighbouring vector pixels

(2) Arrange δ_is in the ascending order. Assign the vector pixel $\mathbf{x}_i$ a rank equal to that of δ_i.

Thus, an ordering $\delta_{(1)} \le \delta_{(2)} \le .... \le \delta_{(N)}$ implies the same ordering of the corresponding vectors given as $\mathbf{x}_{(1)} \le \mathbf{x}_{(2)} \le .... \le \mathbf{x}_{(N)}$

where $\mathbf{x}_{(1)} \le \mathbf{x}_{(2)} \le .... \le \mathbf{x}_{(N)}$ are the rank-ordered vector pixels with the number inside the parentheses denoting the corresponding rank.

Computation of vector median filter (contd..)

The set of rank ordered vector pixels is given by

$$\boldsymbol{X}_{\mathrm{R}} = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \ldots, \boldsymbol{x}_{(N)}\}$$

(3) Take the vector median as $\boldsymbol{x}_{\mathrm{VMF}} = \boldsymbol{x}_{(1)}$

The vector median is defined as the vector that corresponds to the *minimum SOD* to all other vector pixels

(a) (b) (c)

Algorithm of VMF

By definition the VMF of an $n \times m$ RGB image $\boldsymbol{f}$ is computed as follows: Go through the image with a square mask W of size $ms \times ms$. The VMF $\boldsymbol{g}$ of $\boldsymbol{f}$ at the center of the mask (i, j) is

$$g_{i,j} = \boldsymbol{f}(i_{Min}, j_{Min}) \tag{1}$$

where (i_{Min}, j_{Min}) is the position of the pixel with the minimal distance sum to all other pixels within the mask:

$$R_{i_{Min}, j_{Min}} = \min_{(\mu,\nu)\in W} R_{\mu,\nu} \tag{2}$$

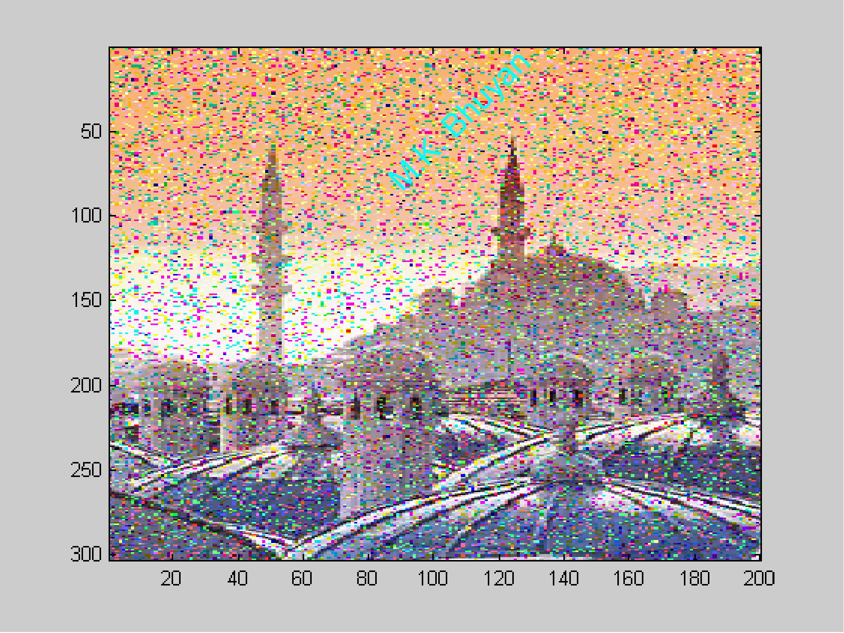
where

$$R_{\mu,\nu} = \sum_{(k,l)\in W} \text{Distance}(\boldsymbol{f}_{\mu,\nu}, \boldsymbol{f}_{k,l}) \; . \tag{3}$$

and $Distance$ is the distance function between two pixels.

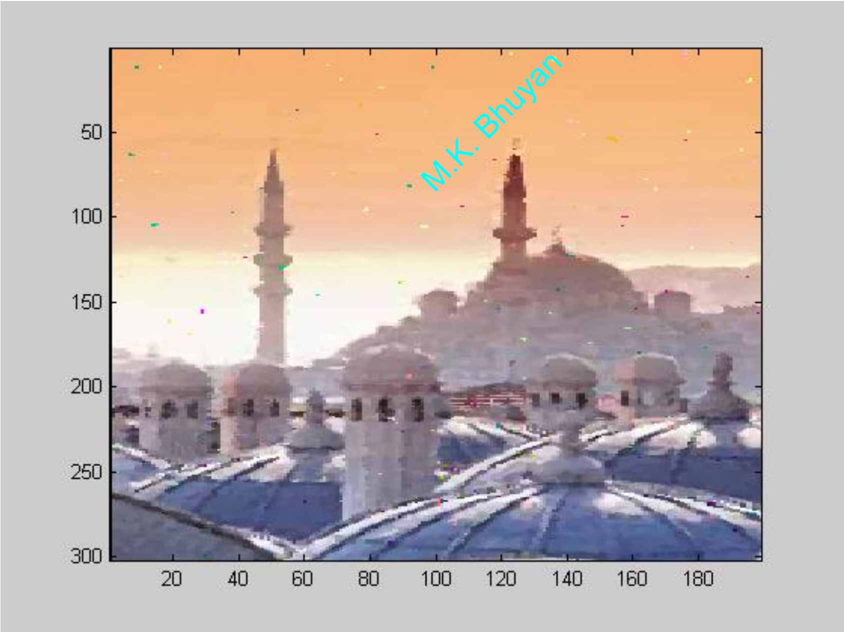
MEDIAN FILTER ON COLOR IMAGES WITH NOISE

Image containing %15 Salt and Pepper Noise



MEDIAN FILTER ON COLOR IMAGES WITH NOISE

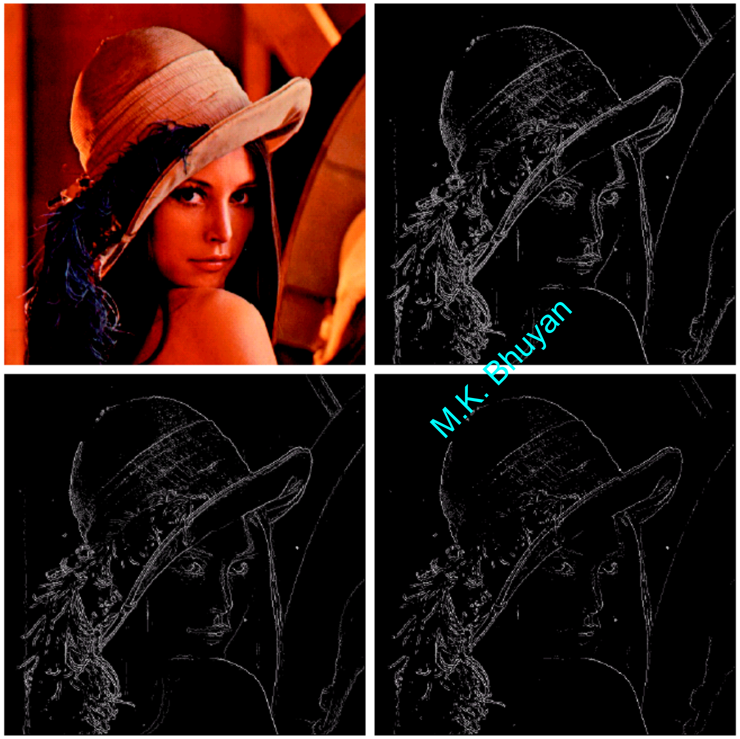
Resulting Image after applying vector median filter



Edge Detection and Colour image segmentation

a b
c d

FIGURE 6.46
(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).



Considering the vector pixels as feature vectors we can apply clustering technique to segment the colour image