IMAGE REPRESENTATION AND ENHANCEMENT

ASSUMING IMAGE IS FORMED

Digital Image Processing

1D to 2D

Image Formation

Image Representation; Enhancement

Image Segmentation

Canny Edge detection

Detectors and Descriptors

Paper Reading; Paper Implementation

Image Transforms

Image Denoising; Compression

Image Formation

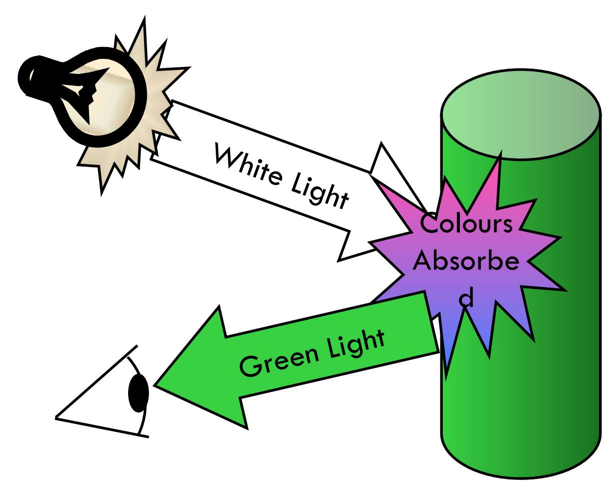
Source of energy → Interaction with 3D world → Image

IMAGE FORMATION

Source of energy [light, ultrasound, electrons]

Geometry [3D world, Camera]

Appearance [Intensity, Color]



Representing Images I(x,y)

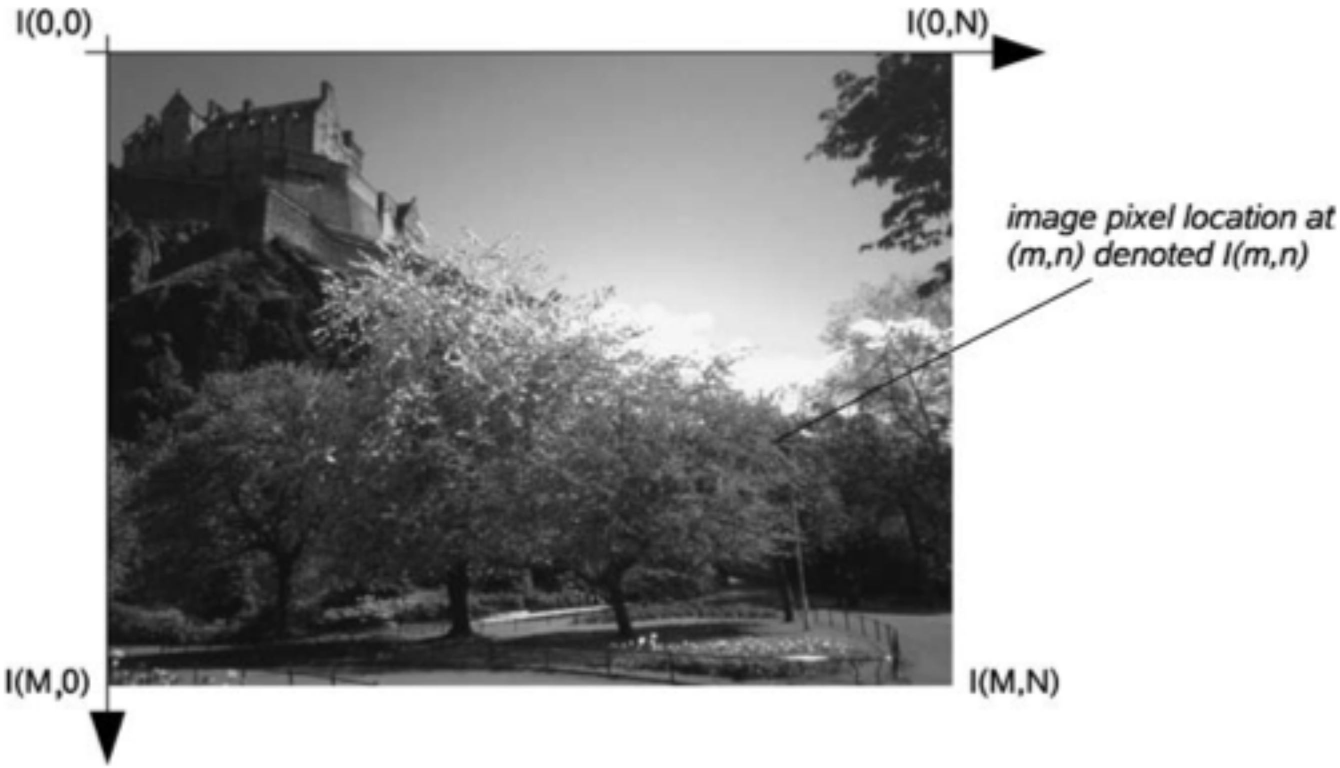
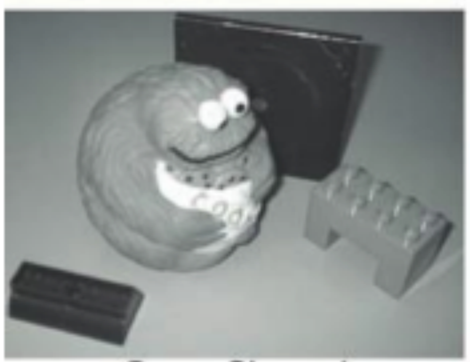


Figure 1.1 The 2-D Cartesian coordinate space of an M x N digital image

Representing Images
R(x,y), G(x,y), B(x,y)

Original

Red Channel



Green Channel

Blue Channel

Representing Images
RGB vs HSV

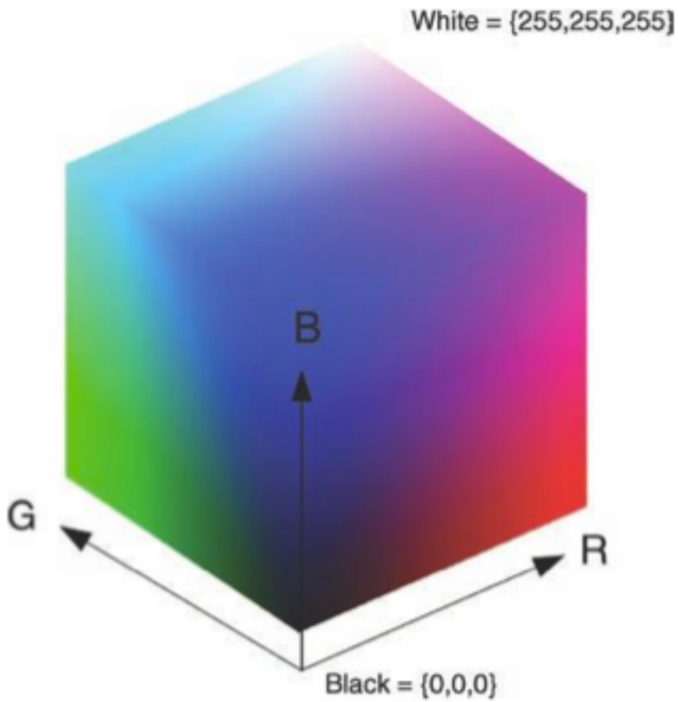


Figure 1.7 An illustration of RGB colour space as a 3-D cube

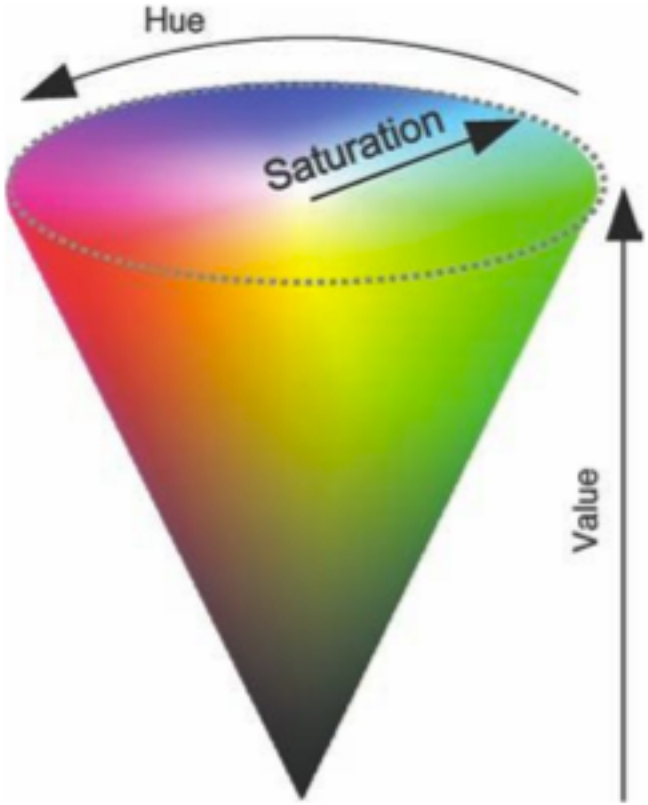


Figure 1.9 HSV colour space as a 3-D cone

Representing Images

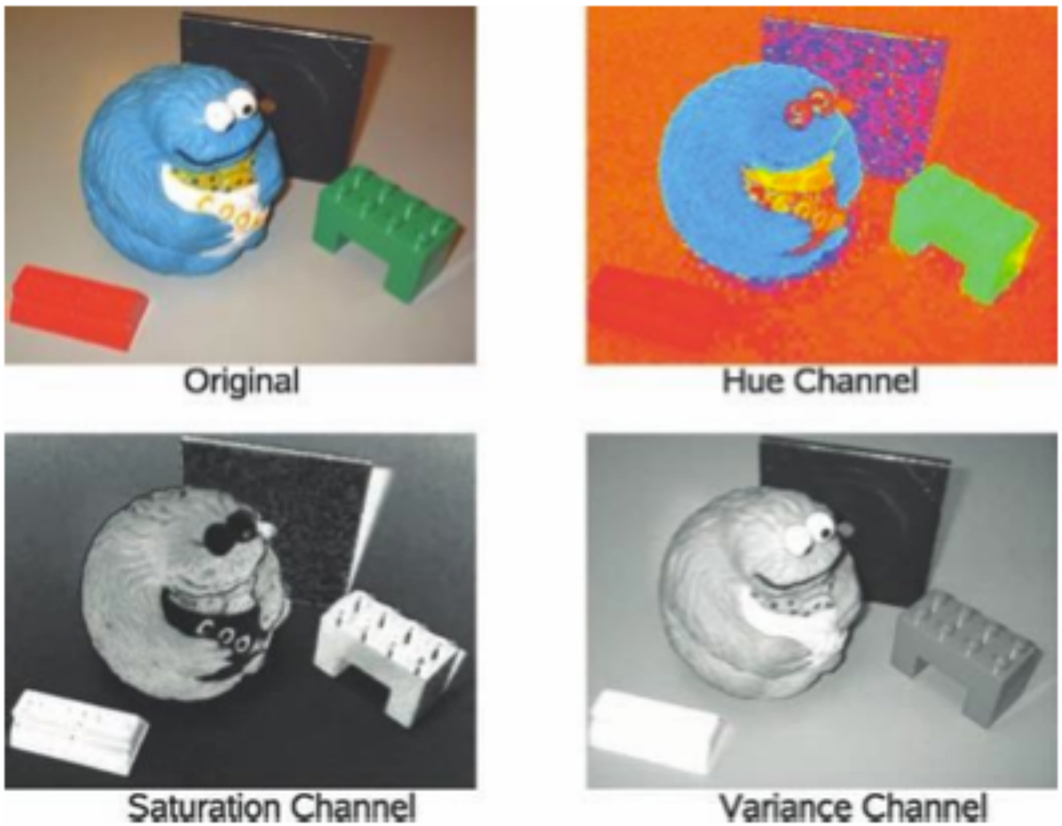


Figure 1.10 Image transformed and displayed in HSV colour space

Representing Images
Color vs Grayscale

Figure 1.8 An example of RGB colour image (left) to grey-scale image (right) conversion

Image Processing: MATLAB code

Example 1.7

Matlab code	What is happening?
D=imread('onion.png');	%Read in 8 bit RGB colour image.
Dred=D(:,:,1);	%Extract red channel (first channel)
Dgreen=D(:,:,2);	%Extract green channel (second channel)
Dblue=D(:,:,3);	%Extract blue channel (third channel)
subplot(2,2,1); imshow(D); axis image;	%Display all in 2×2 plot
subplot(2,2,2); imshow(Dred); title('red');	%Display and label
subplot(2,2,3); imshow(Dgreen); title('green');	
subplot(2,2,4); imshow(Dblue); title('blue');	

Comments

- Note how we can access individual channels of an RGB image and extract them as separate images in their own right.

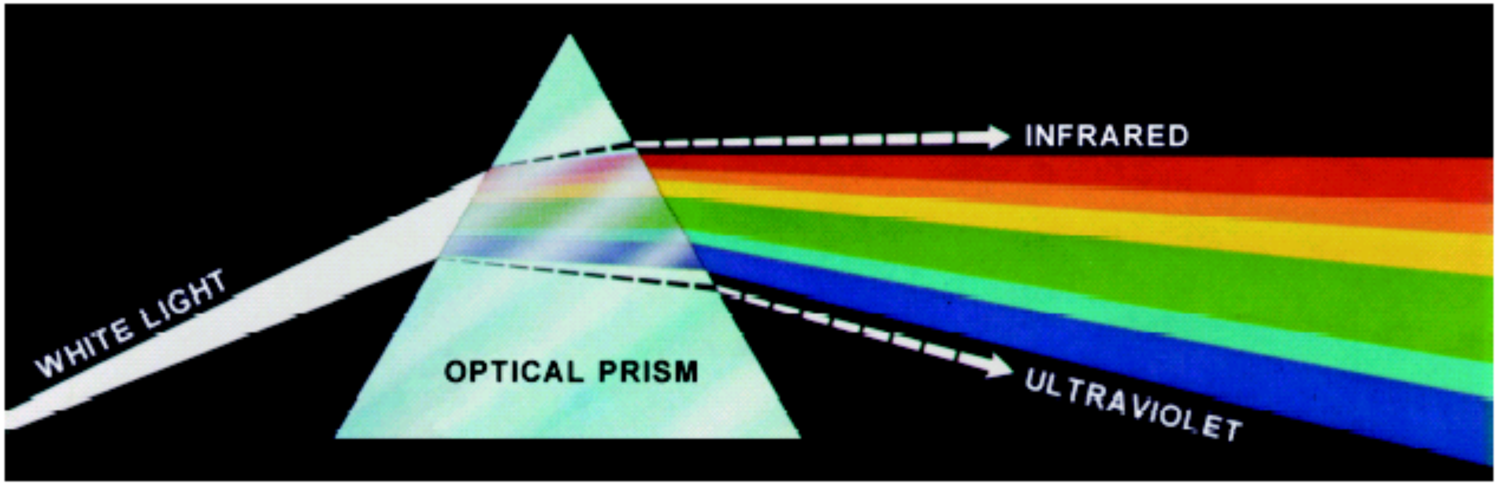
IMAGE REPRESENTATION:
COLOR AND GRAYSCALE

Color

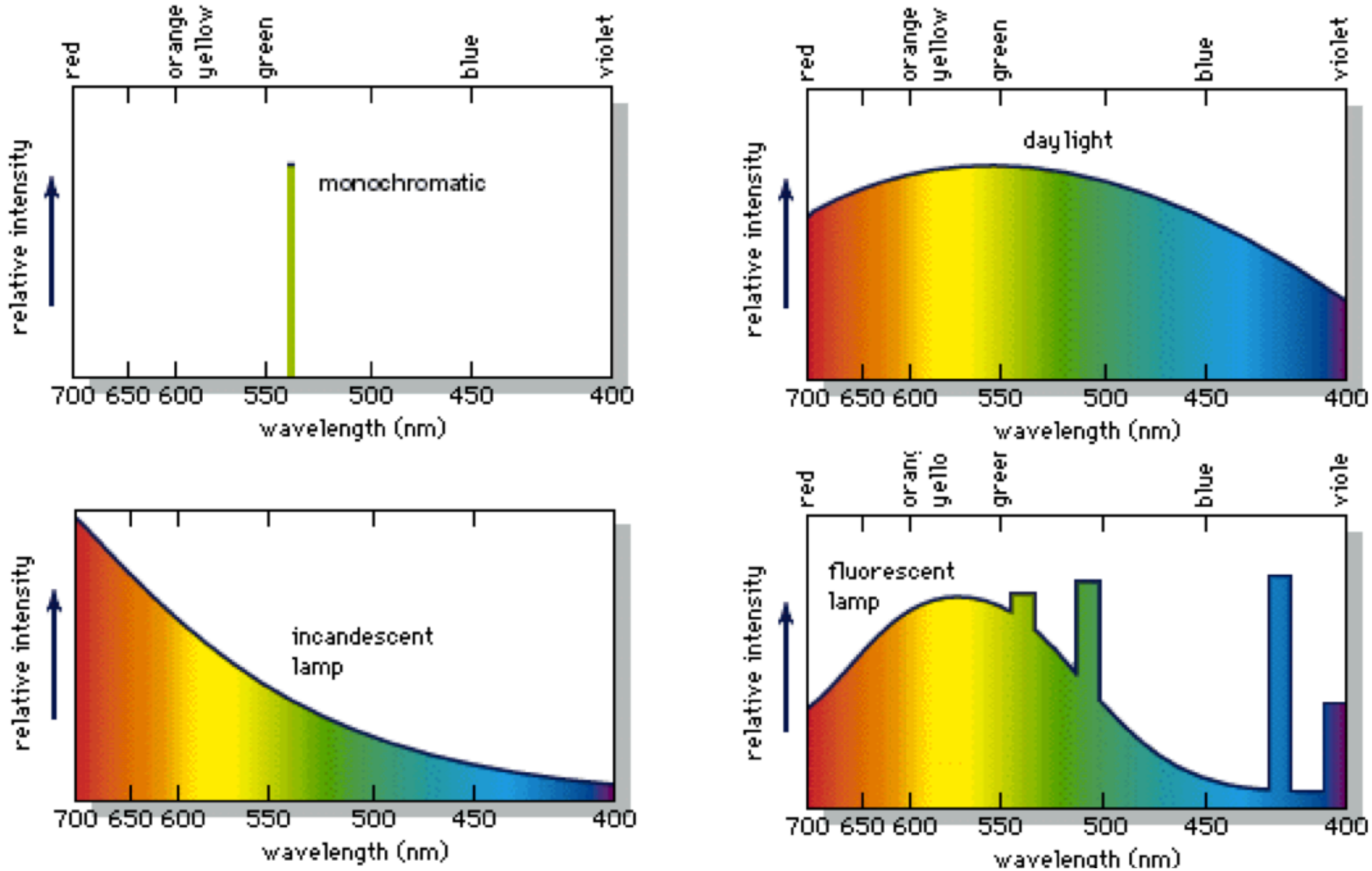
- Can RGB generate all visible colors – color perception?
- How do we agree on a color – color standardization?
- Tease out grayscale/intensity and chroma; move closer to how humans perceive color differences
- Which color representation (and why) is used in
  - TV monitors
  - Printers
  - Computer graphics
  - Image retrieval – for capturing how eyes perceive color differences?

White light

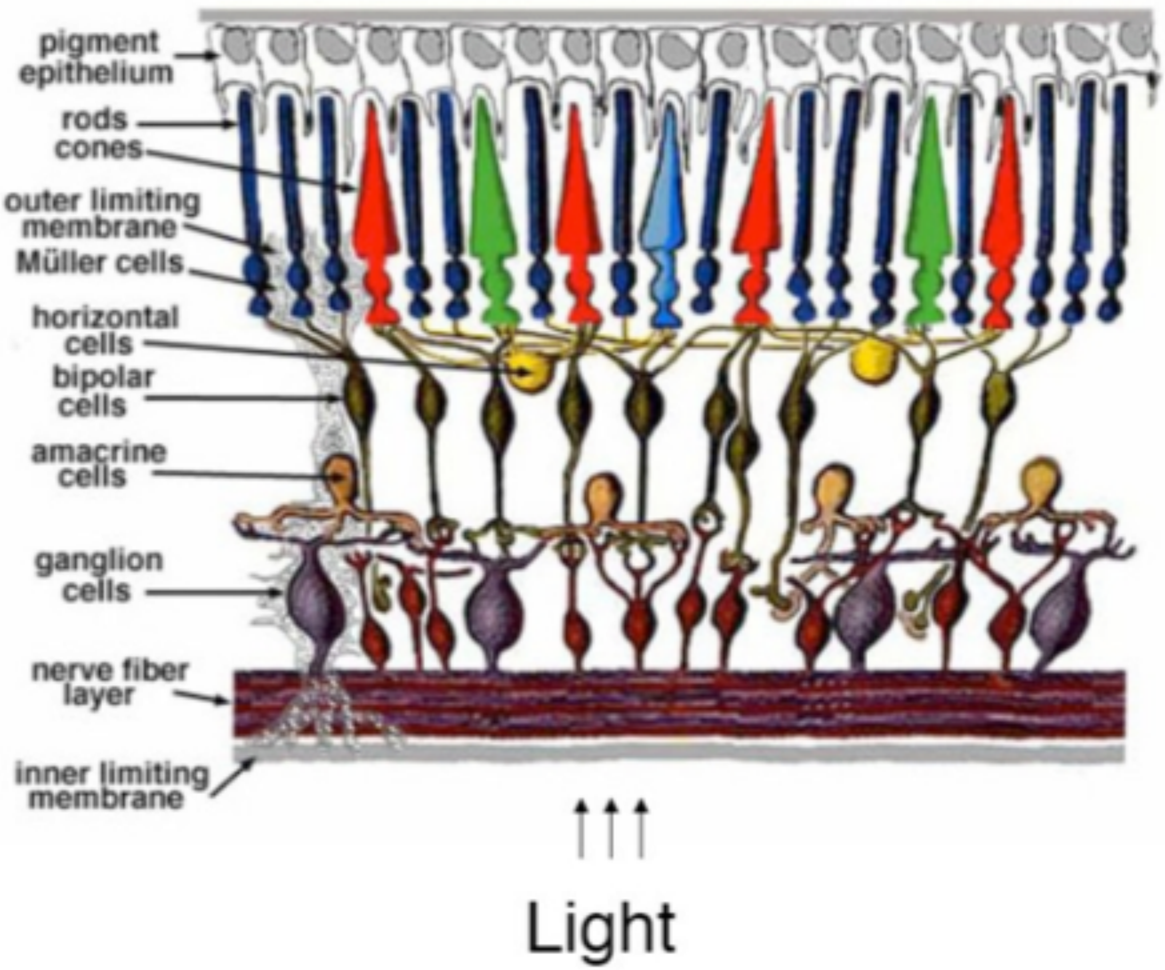
In 1666 Sir Isaac Newton discovered that when a beam of sunlight passes through a glass prism, the emerging beam is split into a spectrum of colors



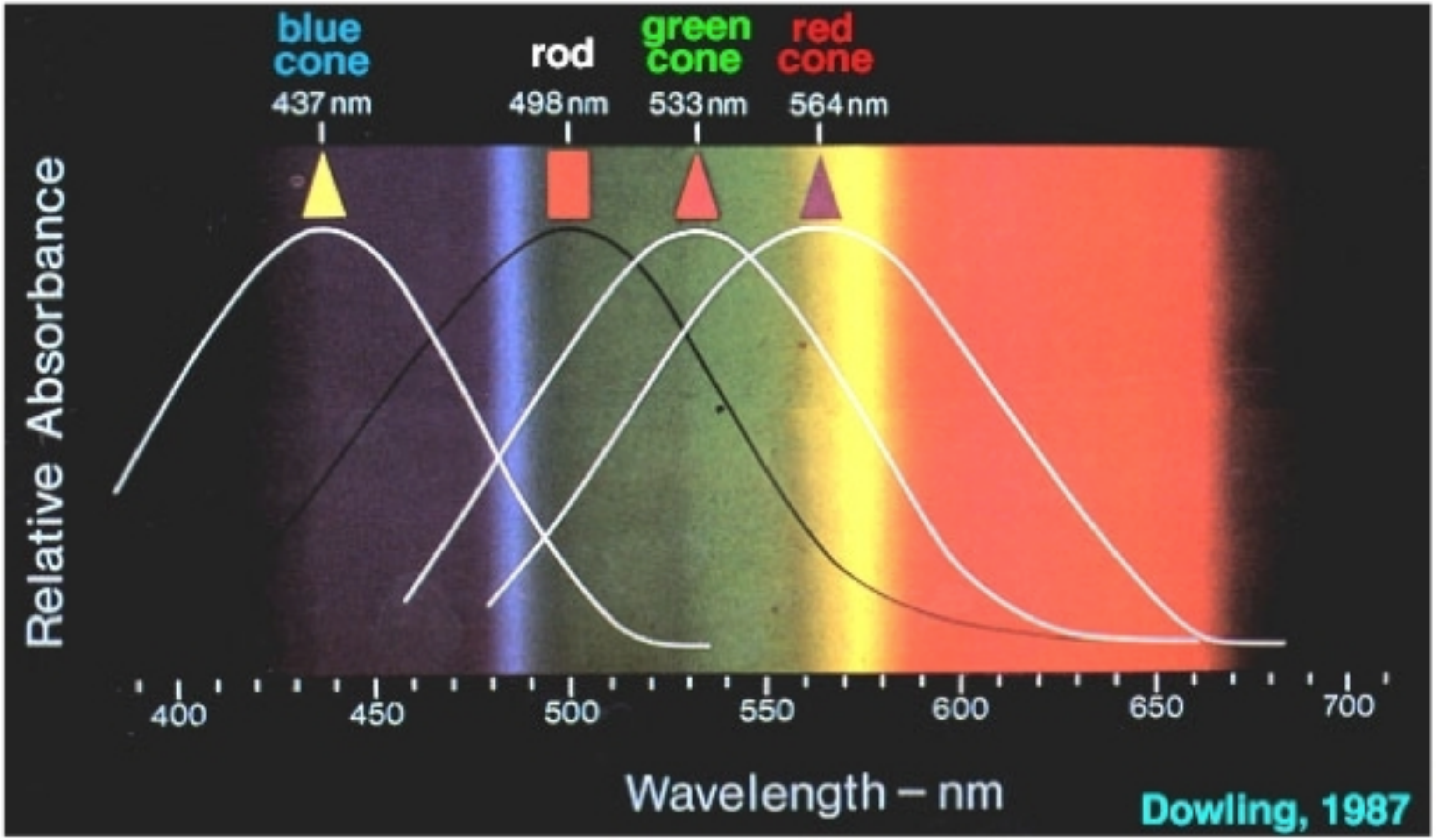
Light Sources



How eyes perceive color

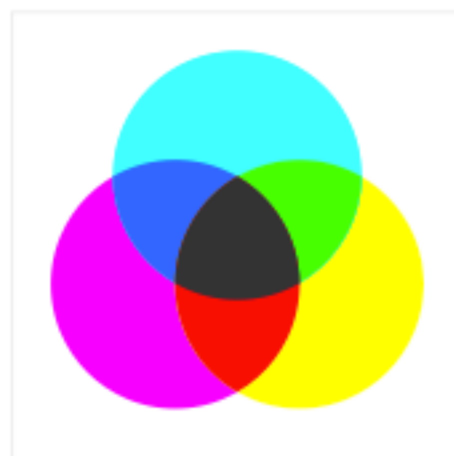


How eyes perceive color

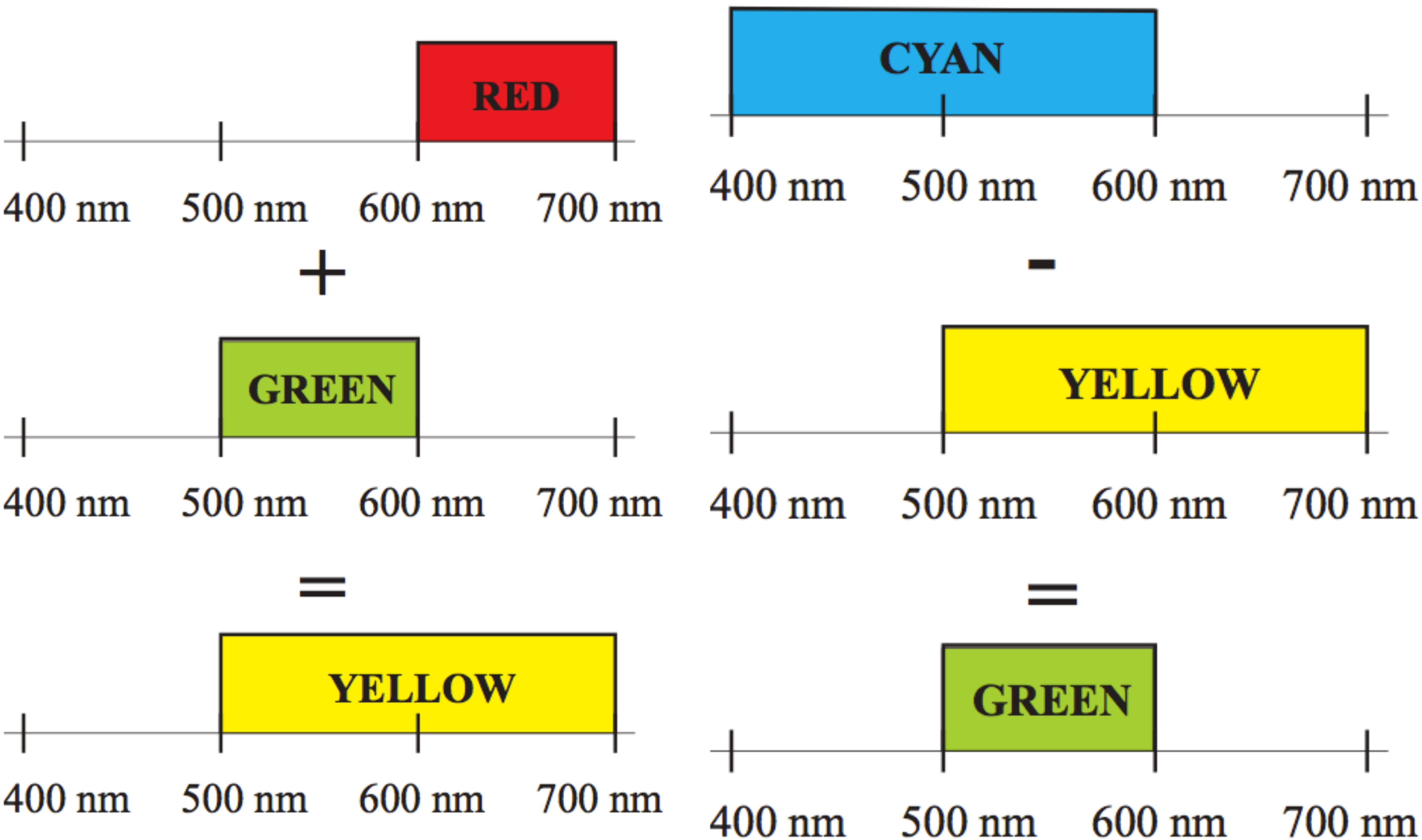


Basics: Color Image Processing

- Additive nature e.g. TV monitors
- Subtractive nature e.g. Printers



Additive vs Subtractive

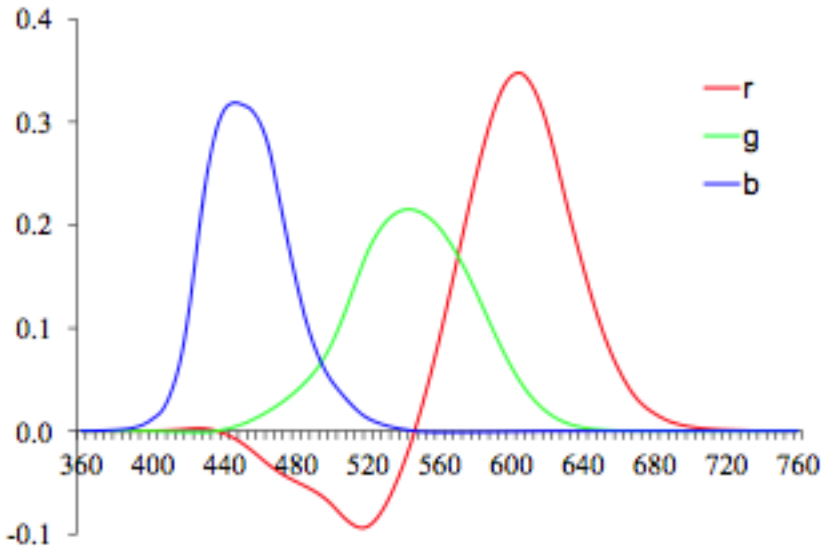


Basics: Color Image Processing

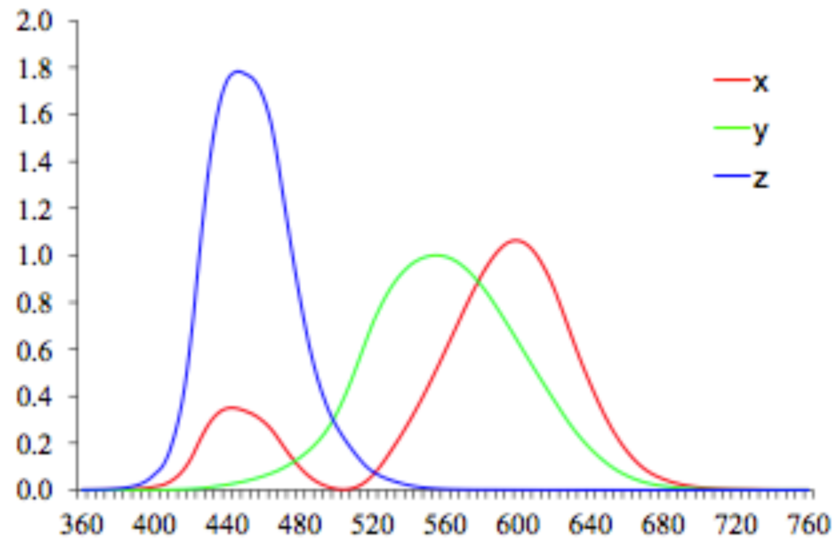
- Can we add deltas in frequency and obtain any other color?
  - Tristimulus nature of our visual systems

CIE RGB and XYZ

- 1930s Commission Internationale d'Eclairage (CIE) by performing color matching experiments
- Standardised RGB representation
- R – 700 nm; G – 546.1 nm; B – 435.8 nm



(a)



(b)

CIE XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

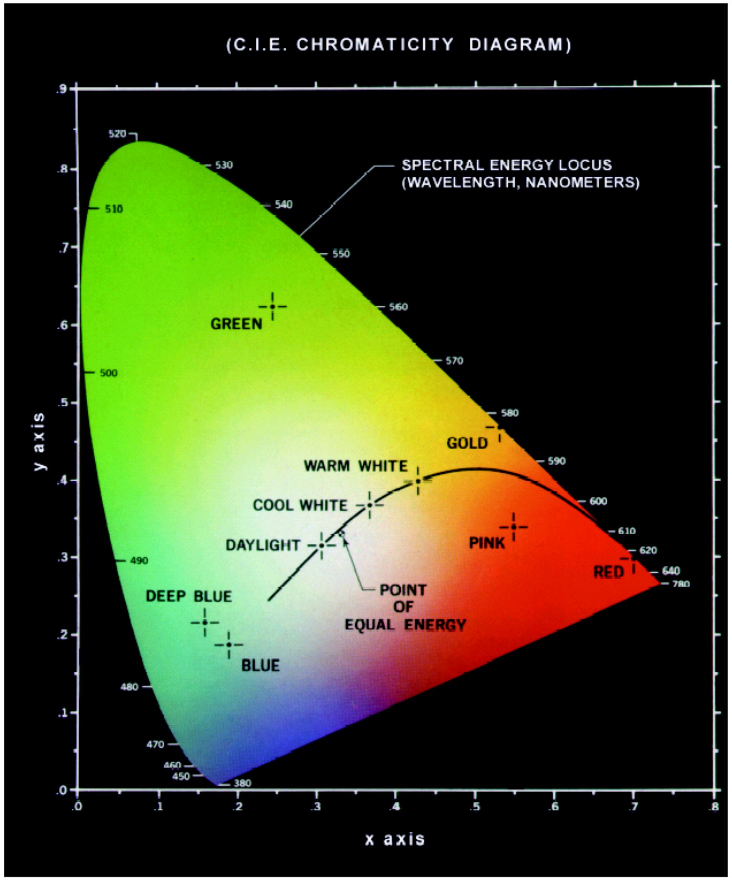
X=Y=Z=1 referent white

Y – Luminance component

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

xyz – chromaticity coordinates

Yxy – Luminance component plus two prominent components of chrominance



Basics: Color Image Processing

- Not only did we get color standardization
- We got closer to how humans perceive color
  - Intensity/Luminance
  - Color

- Let us consider two such representations
  - HSV/HSI
  - $L^*a^*b^*$

RGB vs HSV

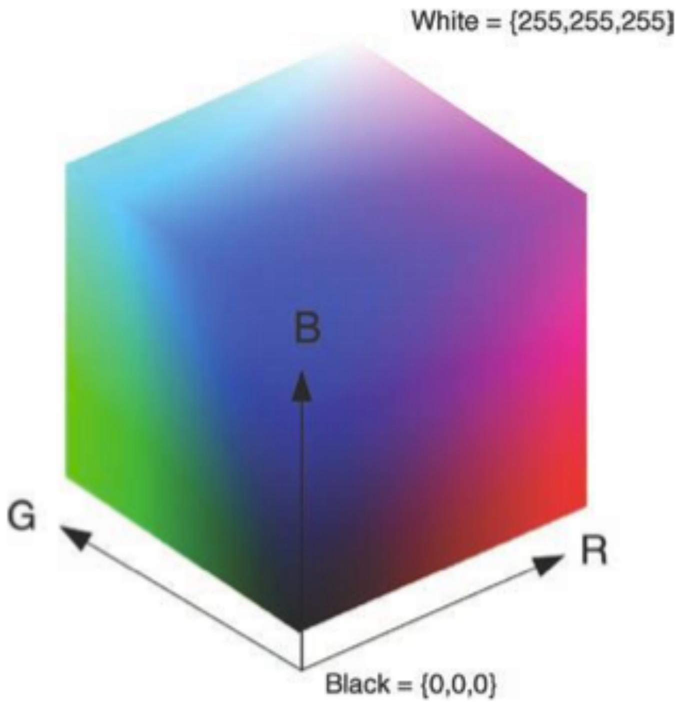


Figure 1.7 An illustration of RGB colour space as a 3-D cube

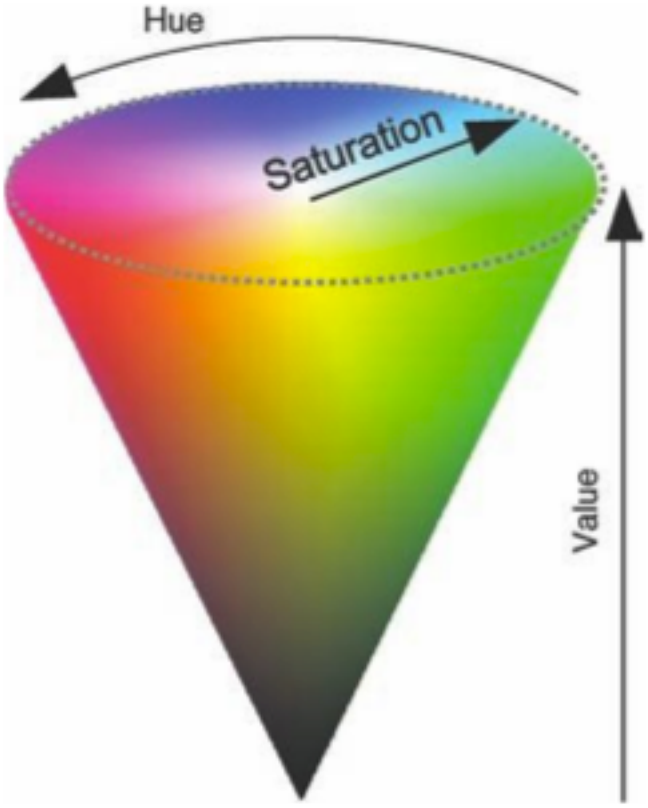
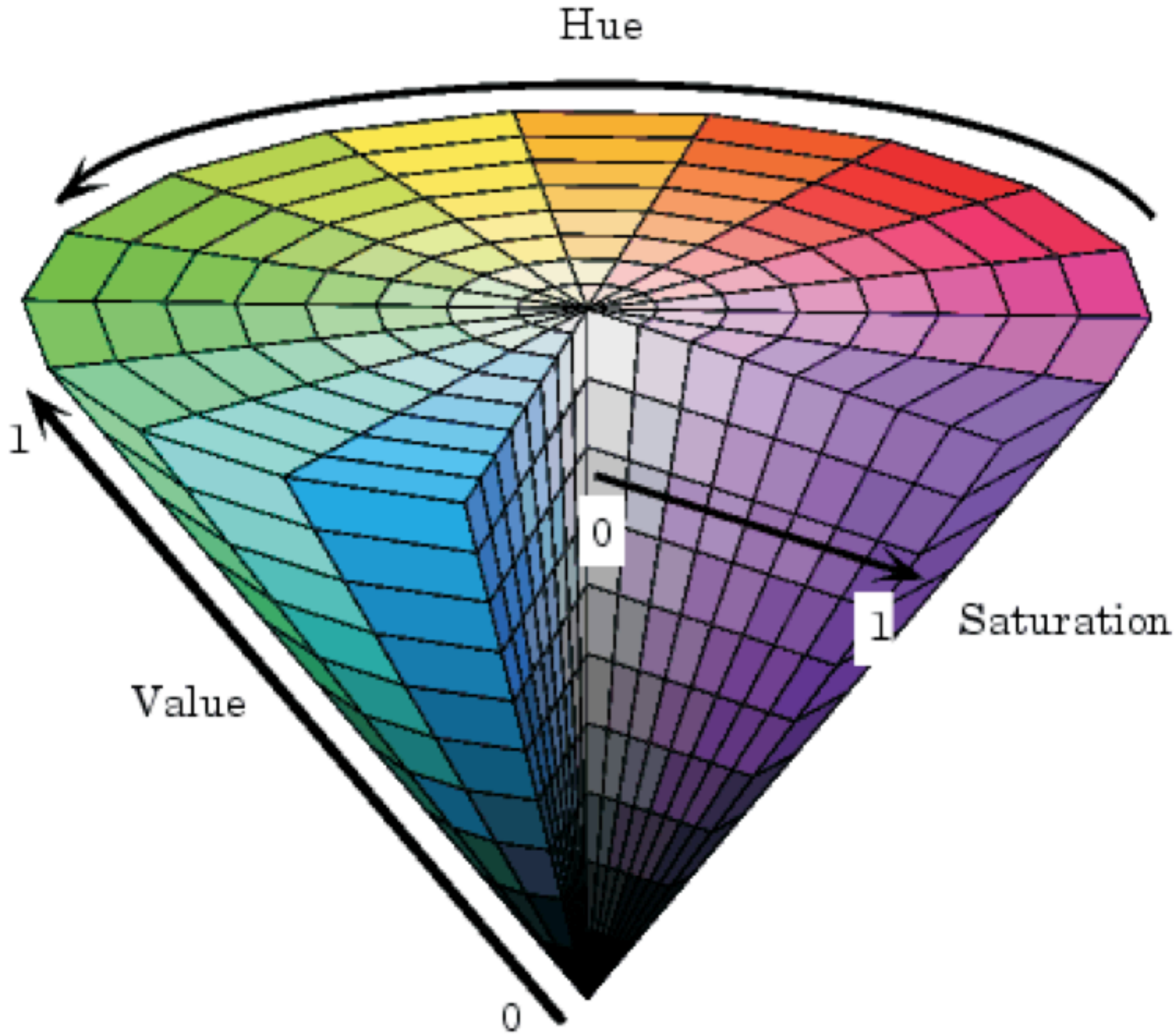


Figure 1.9 HSV colour space as a 3-D cone

HSV

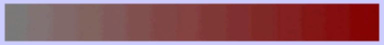


HSI

- *H*=Hue, *S*=Saturation, *I*=intensity
- intensity *I*: 
- H and S may characterize a color: Chromatics

Hue 

- associated with the dominant wavelength in the mixture of light waves, as perceived by an observer.
- is the color attribute that describes a pure color

Saturation 

- relative purity
- amount of white light in the color
- mixed with hue

Example

Pure colors are fully saturated. Not saturated is, e.g., pink (red+white).

L*a*b*

- Based on how humans perceive differences in color
- Perceptually linear – change of same amount of color value produces a change of same visual importance
- CIE defined a non-linear mapping

The L* component of *lightness* is defined as

$$L^* = 116 f\left(\frac{Y}{Y_n}\right),$$

where Y_n is the luminance value for nominal white (Fairchild 2005) and

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases}$$

is a finite-slope approximation to the cube root with $\delta = 6/29$. The resulting $0 \ldots 100$ scale roughly measures equal amounts of lightness perceptibility.

In a similar fashion, the a* and b* components are defined as

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \text{ and } b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

RGB – L*a*b* - HSV

(a) RGB

(b) R

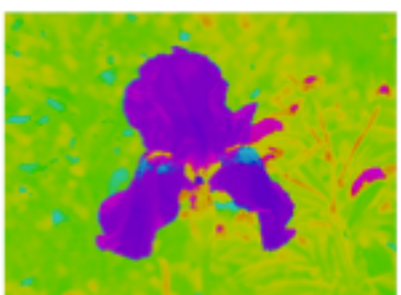
(c) G

(d) B

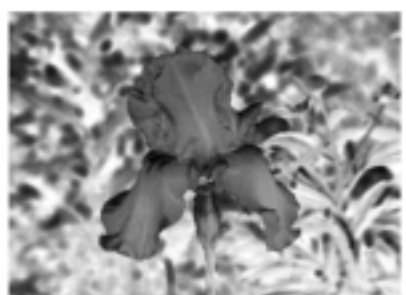
(i) L*

(j) a*

(k) b*



(l) H



(m) S

(n) V

Colorless

Fig. 13a. Color photograph   b. CIELAB L^* c. Rec. 601 luma Y'

d. Component average: "intensity" I e. HSV value V f. HSL lightness L

Colorless

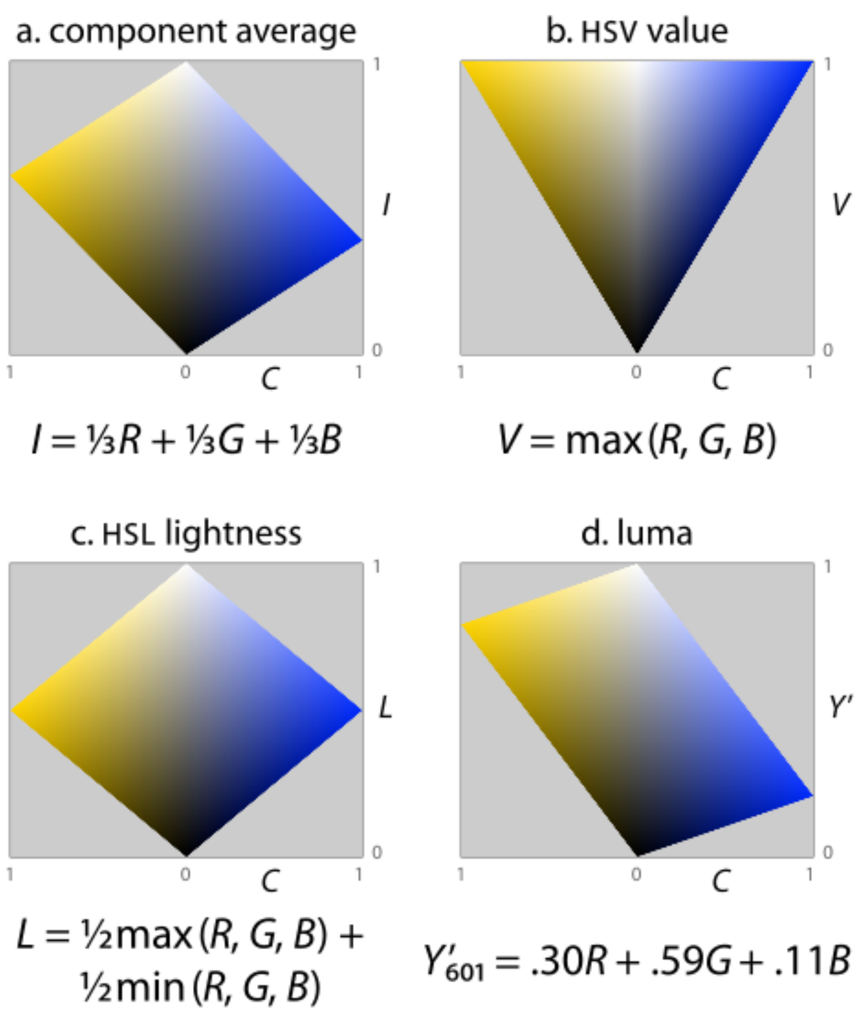

a. component average
b. HSV value
c. HSL lightness
d. luma
I
V
L
Y′
C
0
1
I = ⅓R + ⅓G + ⅓B
V = max (R, G, B)
L = ½max (R, G, B) + ½min (R, G, B)
Y′601 = .30R + .59G + .11B

IMAGE ENHANCEMENT

Overview

- Basic Image Enhancement techniques
- Whitening
- Histogram Equalization
- Smoothening
- Edge detection

- Assignment 1 (LONG)

Image Enhancement: Whitening

$$\mu = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} p_{ij}}{IJ}$$

$$\sigma^2 = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} (p_{ij}-\mu)^2}{IJ}. \tag{13.1}$$

These statistics are used to transform each pixel value separately so that,

$$x_{ij} = \frac{p_{ij}-\mu}{\sigma}. \tag{13.2}$$

Input: p_{ij}

Output: x_{ij}

Image Histograms

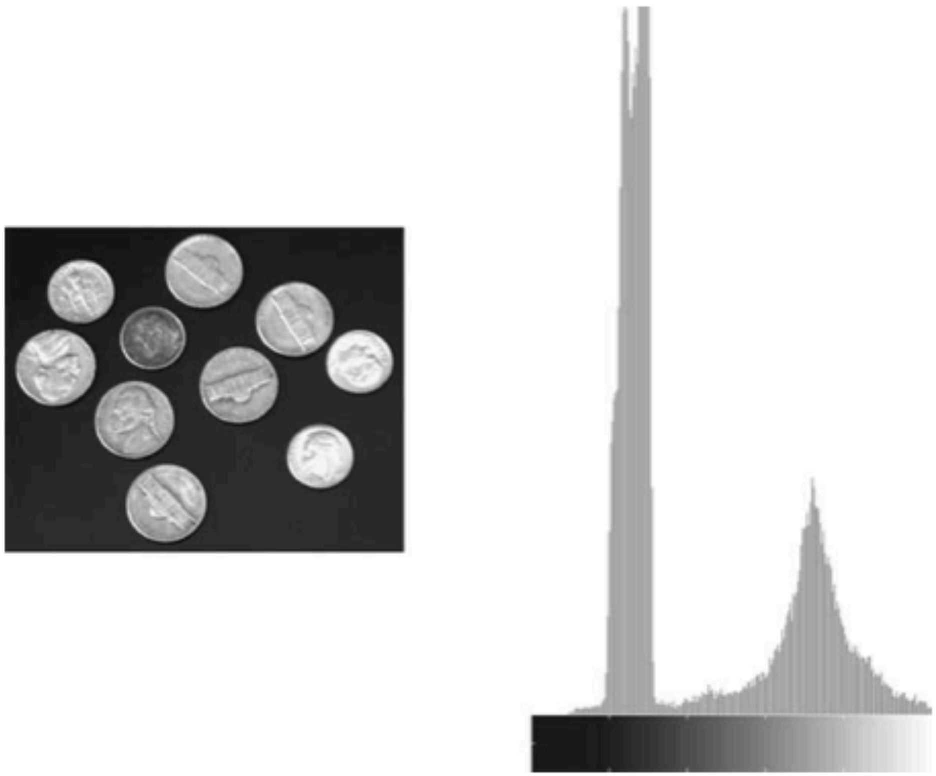


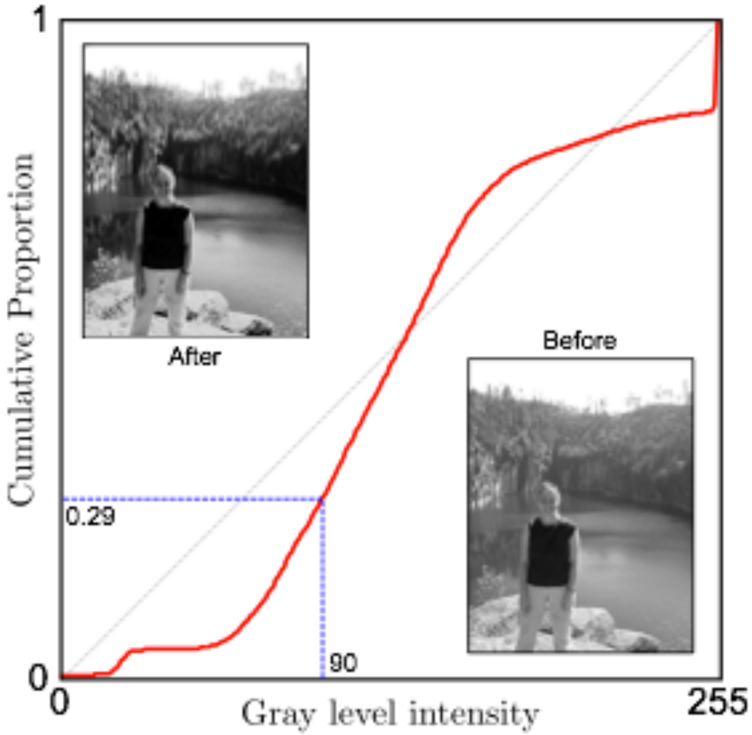
Figure 3.12 Sample image and corresponding image histogram

Image Enhancement: Histogram Equalization

$$h_k = \sum_{i=1}^{I} \sum_{j=1}^{J} \delta[p_{ij} - k],$$

$$c_k = \frac{\sum_{l=1}^{k} h_l}{IJ}$$

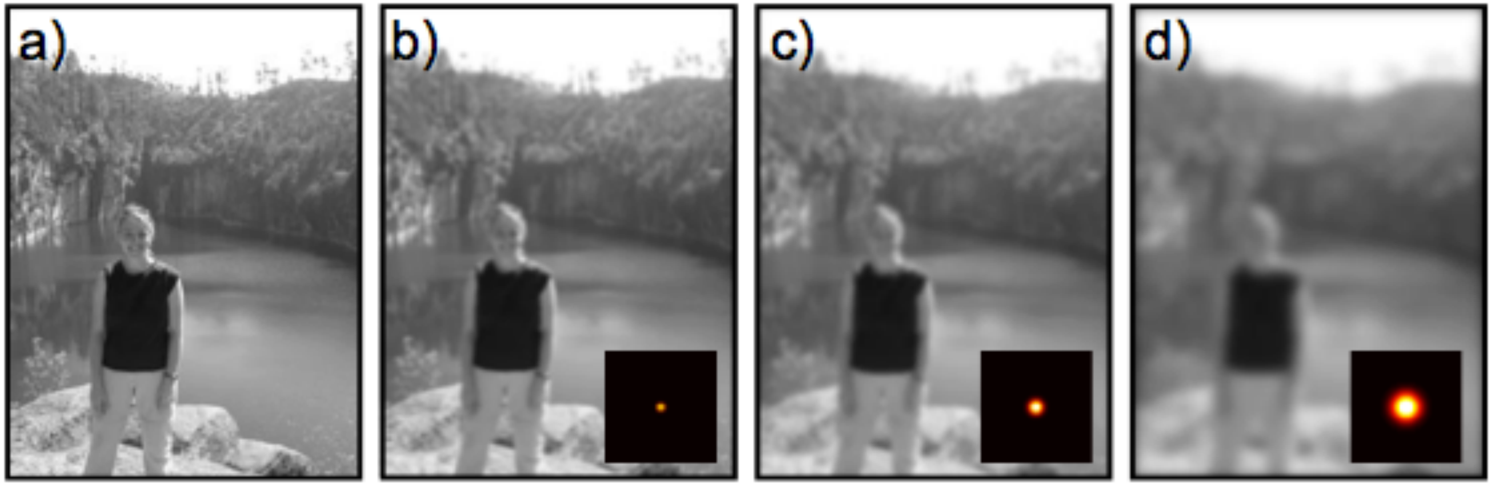
$$x_{ij} = K c_{p_{ij}}$$



90/255 = 0.35
0.29*255 = 74

Image Enhancement:
b. Whitening and c. Histogram Equal.

Image Smoothening / Blurring



$$x_{ij} = \sum_{m=-M}^{M} \sum_{n=-N}^{N} p_{i-m,j-n} f_{m,n}$$

Linear Filtering

$$f(m,n) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{m^2+n^2}{2\sigma^2}\right]$$

Image Smoothening

$$f(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

8	12	8
12	20	12
8	12	8

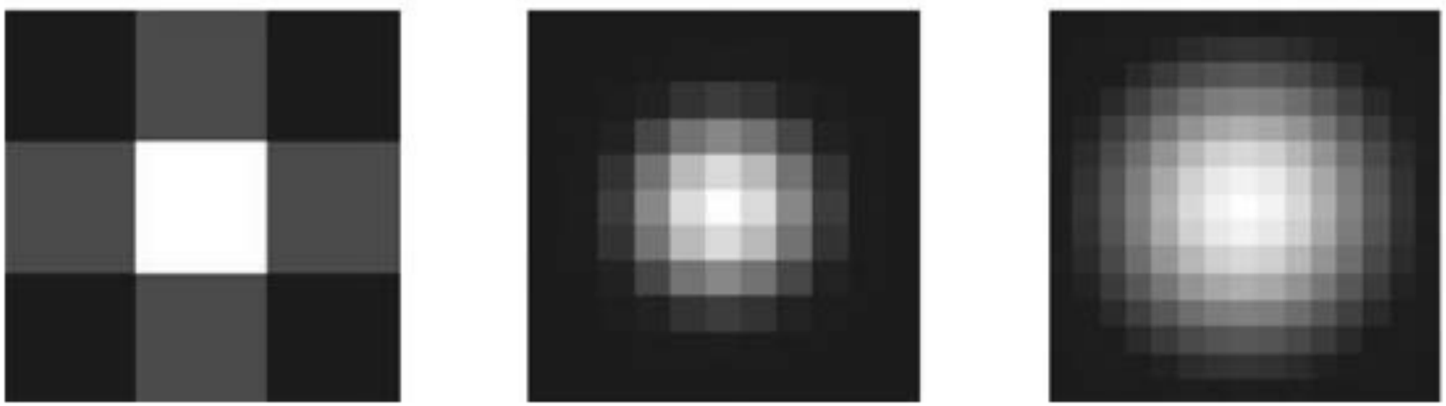


Figure 4.7 Gaussian filter kernels $3\times3\,\sigma=1$, $11\times11\ \sigma=2$ and $21\times21\ \sigma=4$ (The numerical values shown are unnormalised)

Convolution illustration

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

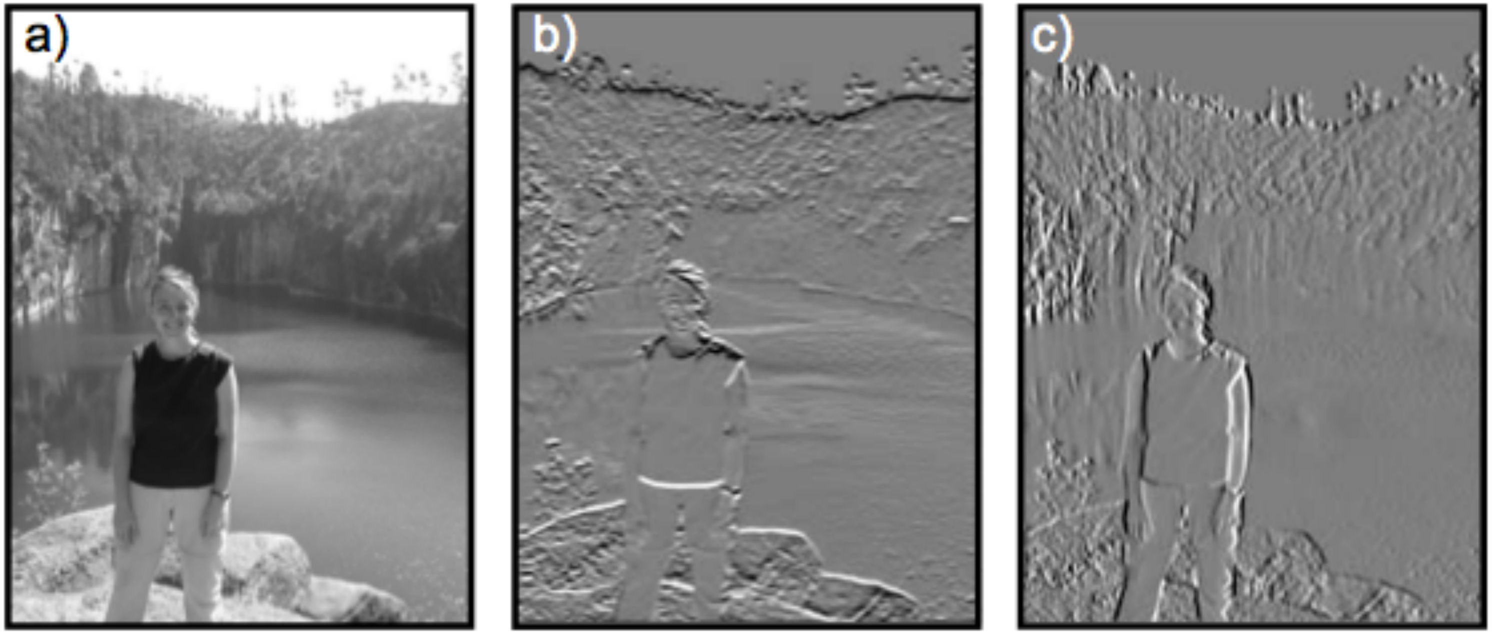
$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = (1*i)+(2*h)+(3*g)+(4*f)+(5*e)+(6*d)+(7*c)+(8*b)+(9*a)$$

Edge detection (first order)

Sobel operators

$$\mathbf{F} = [-1 \;\; 0 \;\; 1]$$

$$\mathbf{F}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \qquad \mathbf{F}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



Original image

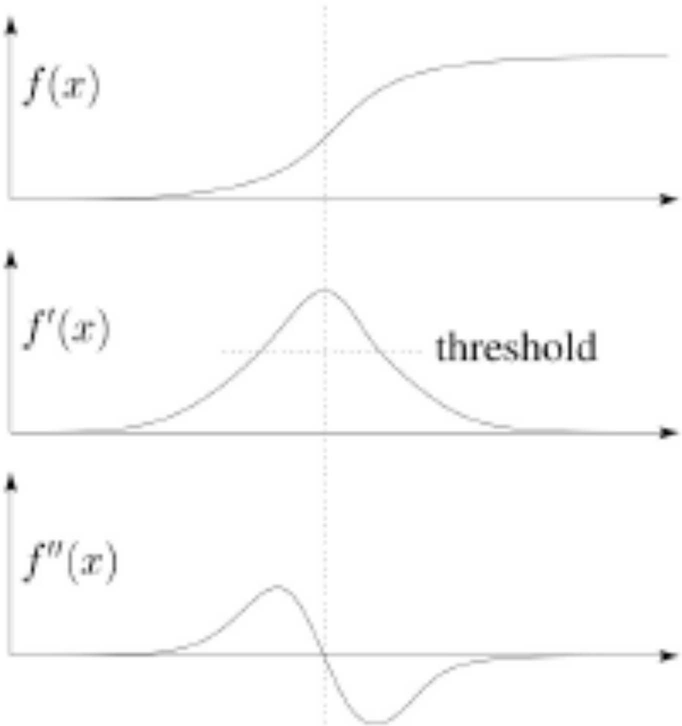
Prewitt (vertical)

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Prewitt (horizontal)

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Second derivative operators



The Sobel operator can produce thick edges.
Ideally, we're looking for infinitely thin boundaries.

An alternative approach is to look for local extrema in the first derivative: places where the change in the gradient is highest.

Comparing first and second order

Original Image

x-derivative
(Sobel)

y-derivative
(Sobel)

$$|G| = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) + \frac{1}{4}\pi$$

Gradient Magnitude

Laplacian
(original kernel)

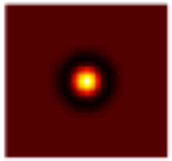
First order – thick edges; Laplacian – thin edges
Less sensitive to noise as compared to Laplacian

Image Smoothening then Edge det.

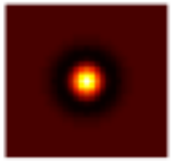
Laplacian

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

Laplacian of Gaussian



Difference of Gaussians



$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\nabla^2 f = f(x+1,y) + f(x-1,y) - 4f(x,y) + f(x,y+1) + f(x,y-1)$$

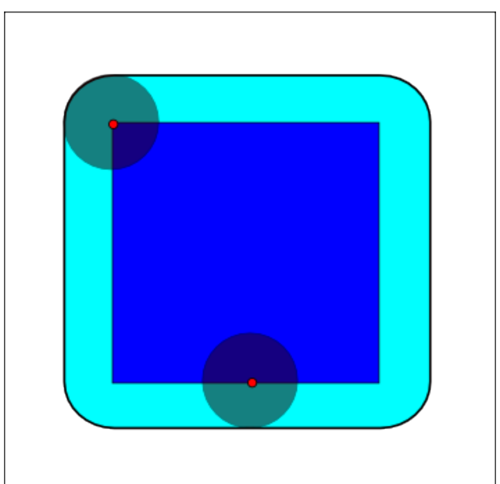
MATLAB code

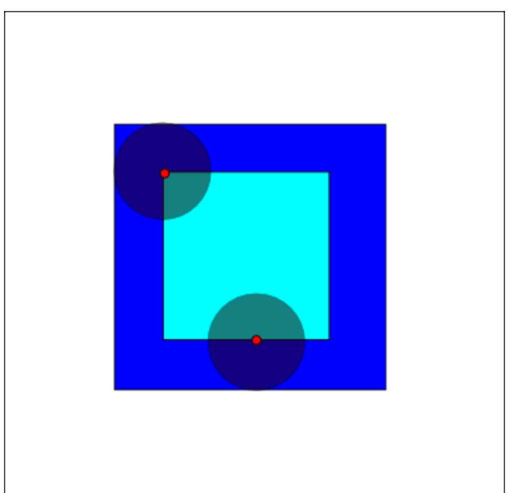
```
a=rgb2gray(a);
a=im2double(a);

figure; colormap('gray'); imagesc(a);

%prewitt
%horizontal - finds horizontal discontinuity (derivative) - vertical edges
dx = [-1 0 1; -1 0 1; -1 0 1];
y = conv2(a, dx);
figure; colormap('gray'); imagesc(y);

%vertical - finds vertical discontinuity (derivative) - horizontal edges
dy = [1 1 1; 0 0 0; -1 -1 -1];
z = conv2(a, dy);
figure; colormap('gray'); imagesc(z);

%gaussian smoothing
dy2 = [1/16 1/8 1/16; 1/8 1/4 1/8; 1/16 1/8 1/16];
z2 = conv2(a, dy2);
figure; colormap('gray'); imagesc(z2);

BW1 = edge(a,'prewitt');
BW2 = edge(a,'canny');
figure; colormap('gray'); imagesc(BW1);
figure; colormap('gray'); imagesc(BW2);
```

Morphological operators

(a) (b) (c) (d) (e) (f)

Figure 3.21 Binary image morphology: (a) original image; (b) dilation; (c) erosion; (d) majority; (e) opening; (f) closing. The structuring element for all examples is a 5×5 square. The effects of majority are a subtle rounding of sharp corners. Opening fails to eliminate the dot, since it is not wide enough.





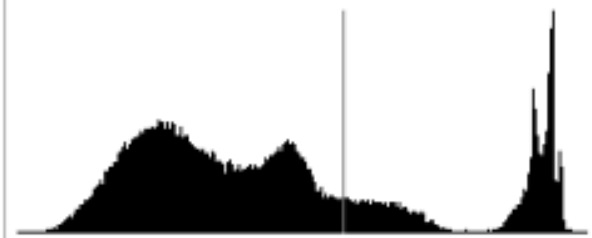
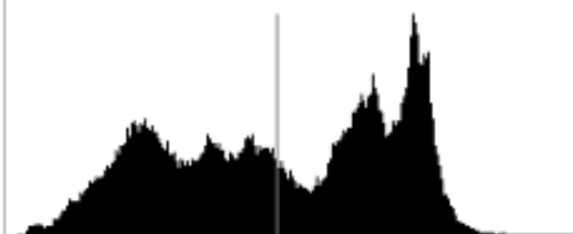
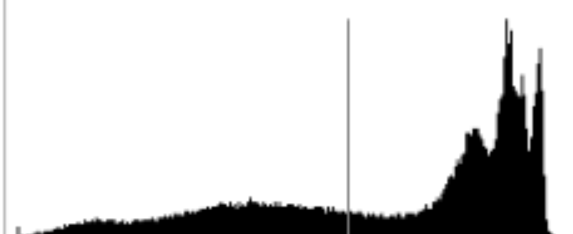
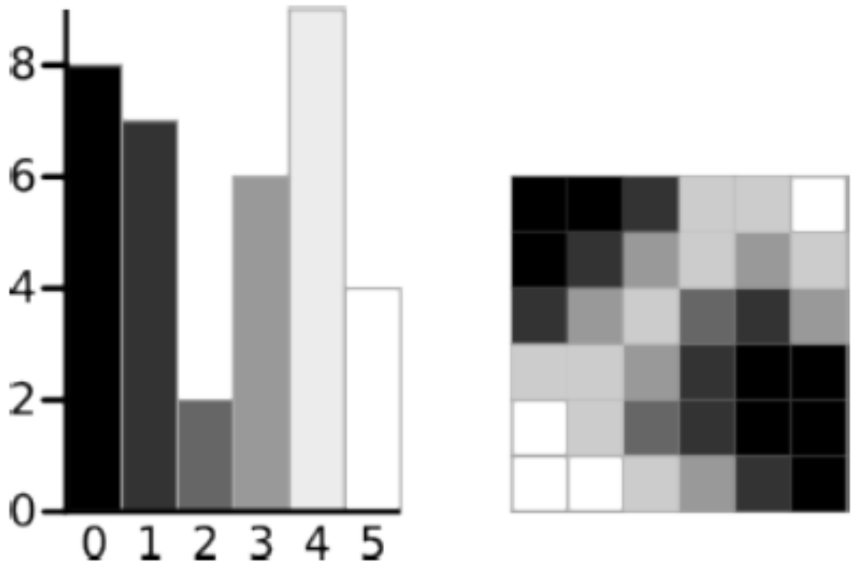
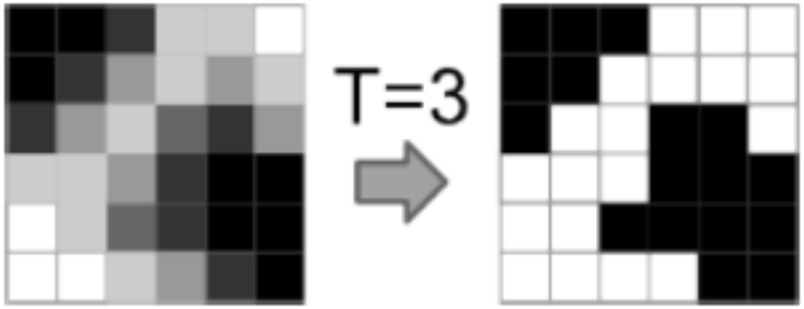
Greyscale Image	Binary Image	Histogram
		
		
		

Image Segmentation: Otsu's method



A 6-level greyscale image and its histogram



Result of Otsu's Method

Conclusion

- Color representation and models
  - RGB, HSV, $L^*a^*b^*$ (YCbCr)
  - Szeliski (Computer Vision)
  - http://en.wikipedia.org/wiki/HSL_and_HSV

- Image enhancement techniques
  - Prince – Computer Vision (Chapter 13)
  - Best learnt by playing with it

    => Assignment 1(due 21 Jan mid night)

- Acknowledgement: Internet pictures

ASSIGNMENT 1(LMS)

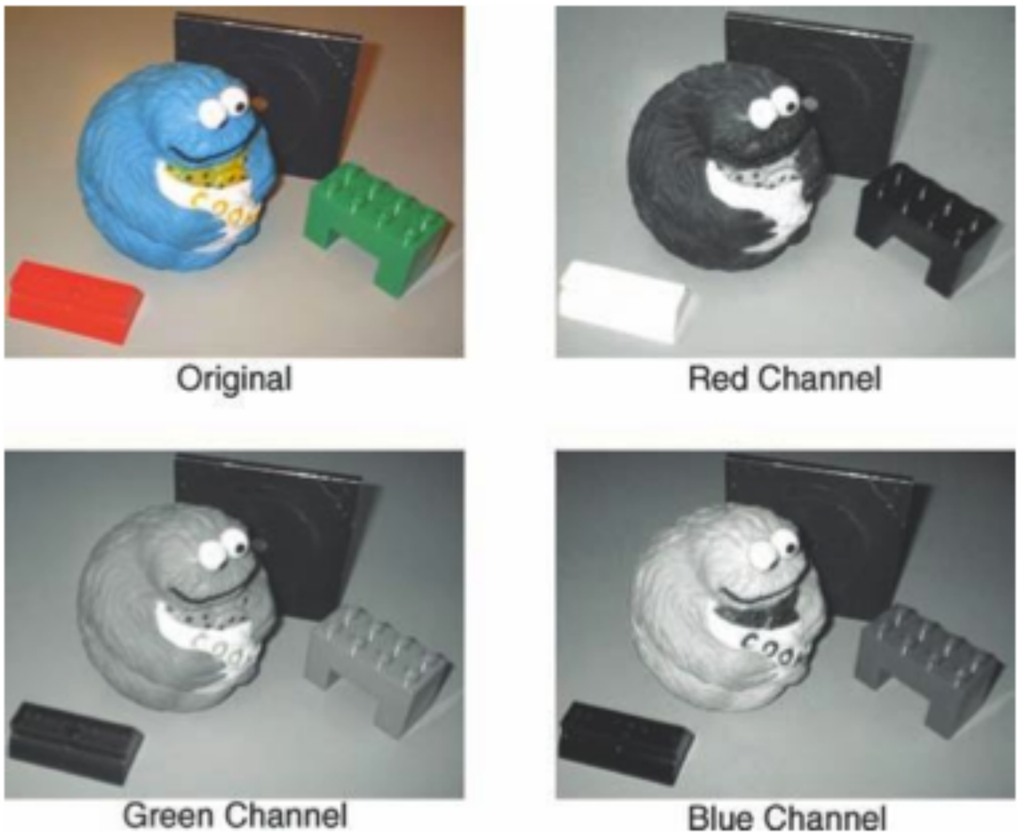
WRITE CLEAR COMMENTS AND OBSERVATIONS

SUBMIT A ZIP FILE WITH PDF, OPENCV CODE AS WELL

10 marks (weightage 0 to 5)

Question 1

- Choose an RGB image (Image1); Plot R, G, and B separately (Write clear comments and observations)



Question 2

- Convert Image 1 into HSL and HSV. Write the expressions for computing H, S and V/I.
- (Write clear comments and observations)

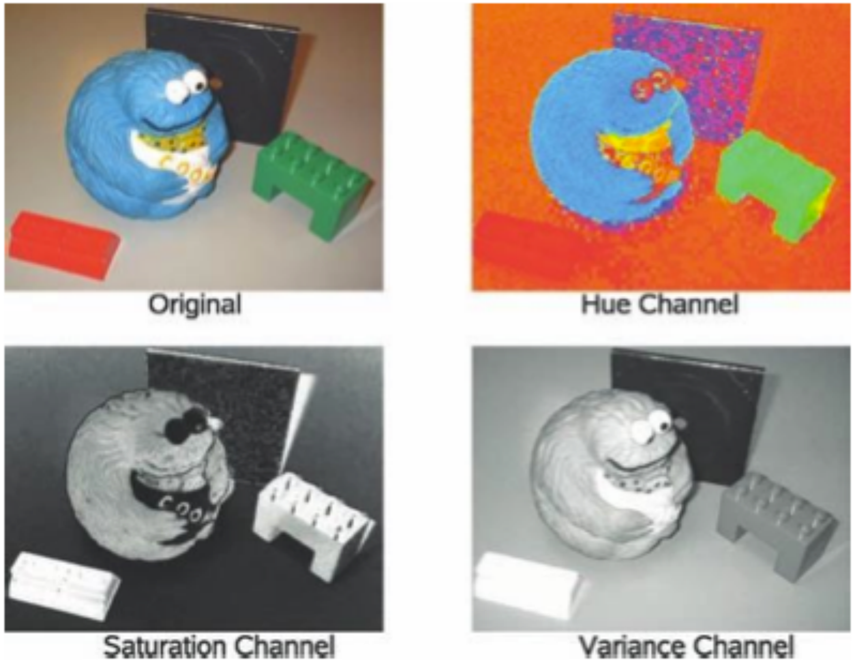


Figure 1.10 Image transformed and displayed in HSV colour space

Question 3

- Convert Image 1 into L*a*b* and plot

Question 4

- Convert Image 1 into Grayscale using the default OpenCV function. Write the expressions used for the conversion.

Figure 1.8 An example of RGB colour image (left) to grey-scale image (right) conversion

Question 5

- Take Image 2 (a selfie of yourself) and implement a skin color detector i.e segment only skin pixels. [Choose any method you think is appropriate]. Describe or Illustrate when your detector will work and when it will fail.

Question 6

- Try some color manipulation
  (mainly color transformation, not mere color substitution)
- [Make yourself fairer or darker]

Question 7

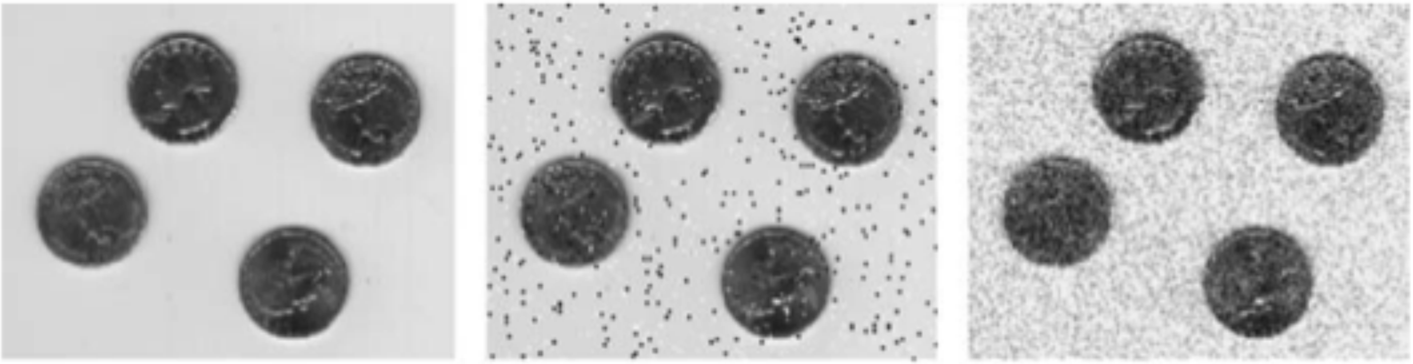
- Take a grayscale image (Image 3) and illustrate
  - Whitening
  - Histogram equalization

Question 8

- Take a low illumination noisy image (Image 4), and perform Gaussian smoothing at different scales. What do you observe w.r.t scale variation?

Question 9

- Take an image (Image 5) and add salt-and-pepper noise. Then perform median filtering to remove this noise.



(a) (b) (c)

Figure 4.3 (a) Original image with (b) 'salt and pepper' noise and (c) Gaussian noise added

Question 10

- Create binary synthetic images to illustrate the effect of Prewitt (both vertical and horizontal) plus sobel operators (both vertical and horizontal)
  - Clue: check when you have a vertical/horizontal strip of white pixels – vary width of the strip from 1 pixel to 5 pixels
  - What do you observe?

Question 11

- What filter will you use to detect a strip of 45 degrees

Question 12

- Take an image and observe the effect of Laplacian filtering
- Can you show edge sharpening using Laplacian edges

Figure 4.14 Edge sharpening using the Laplacian operator

Question 13

- Take an image and show that applying
  - Laplacian after Gaussian filtering
  - Gaussian filtering after Laplacian
- results in similar images

Question 14

- Implement a bounding box detector of number plates of a car. Make sure the method works on 5 different cars. You are free to make some assumptions on the size of the number plate and use any image enhancement techniques along with morphological operations