Lecture-3

Intensity Transformation function:

* Linear
* Logarithmic
* Power law

Image Enhancement methods

→ Spatial Domain: Direct manipulation of pixels in an image.

→ Frequency Domain: Process the image by modifying the Fourier transform of an image.

→ Combination Methods:

(*) Purpose: ① Highlighting interesting detail in Images.

② Removing noise from image

③ Making images more visually appealing.

$$\underbrace{g(x,y)}_{s}^{\text{output}} = T\,\underbrace{[f(x,y)]}_{r}^{\text{input}}$$

T = Gray level / Intensity transformation / mapping funct.

$s = T(r)$

s = Gray level of g at (x, y)

r = Grey level of f at (x, y)

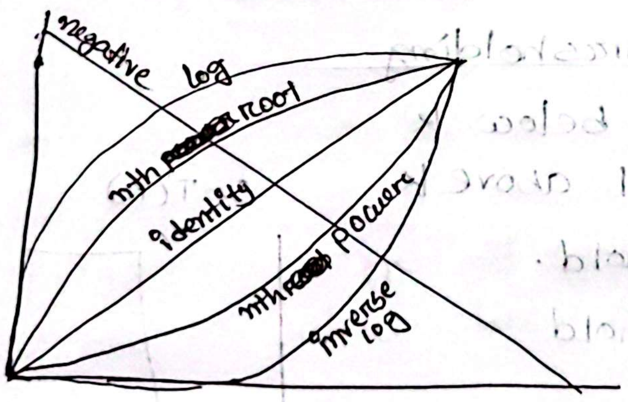
T = is a function that maps r to s

Intensity transformation functions are 2:

① Basic intensity
- → Linear function → Negative Transformation; Identity "
- → logarithmic function → log Transformation; Inverse log "
- → Power-law function → nth power trans..; nth root "

② Piecewise Linear Transformation
- → Contrast streetching, Thresholding
- → Gray level slicing
- → Bit plane slicing.

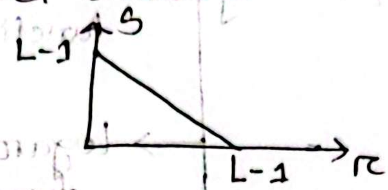
①



Linear (Negative)

1. Reverse the Gray level order.

$s = (L-1) - r$



⊛ Negative images are useful for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size.

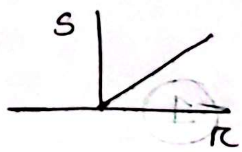
$\boxed{s = 255 - r}$ $L = 2^8 = 256$

Identity

$s = r$

$r_1 = s_1$

$r_2 = s_2$.



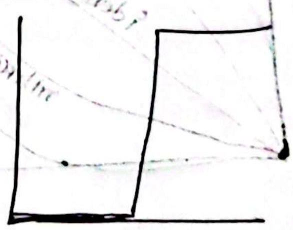
Thresholding

⊛ Darkens level below k

⊛ Brightens level above k

$s = T(r)$

$$s = \begin{cases} 1 & r > \text{Threshold}. \\ 0 & r < \text{threshold} \end{cases}$$



Dark ⟷ light.

(r_1, s_1) (r_2, s_2)

$(r_1, s_1) = (k, 0)$

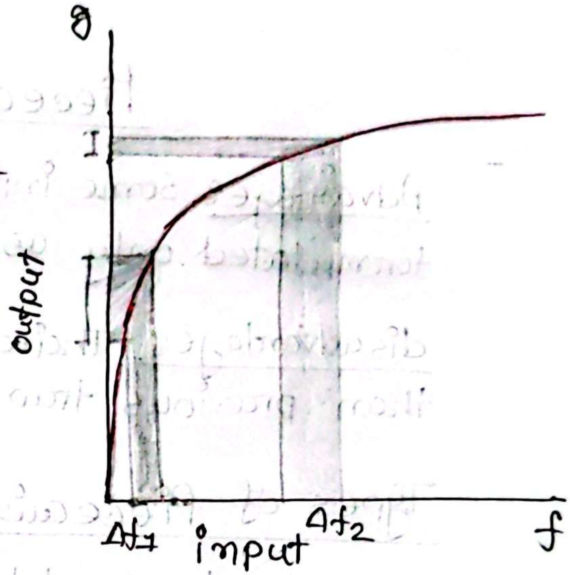
$(r_2, s_2) = (k, L-1)$

Log function

$s = c \times \log(1+r)$

→ maps narrow range of low input values → into a wider range of output values.

→ Inverse log does opposite.



Properties:

1. For lower amplitudes of input image the range of Gray level is expanded.

2. For higher amplitudes of input image the range of Gray levels is compressed.

Also called dynamic-range compression/Expansion.

Power law function

$s = c \times r^{\gamma}$

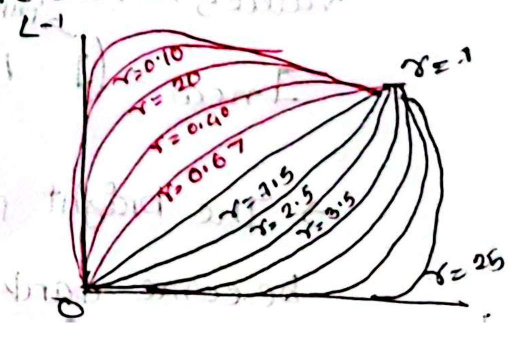
→ map a narrow range of dark input values into a wider range of output values or vice versa.

→ Varying γ gives a whole family of curves.

$\gamma > 1 \rightarrow$ mapping is towards darker output values.

$\gamma = 1 \rightarrow$ Linear (identical)

$\gamma < 1 \rightarrow$ toward brighter output



Gamma Correction

Varying Gamma try to reproduce colors accurately also requires gamma correction.

Piecewise

<u>Advantages</u>: some important transformations can be formulated only as a piecewise function.

<u>disadvantages</u>: Their specification requires more user input than previous transformation.

Types of Piecewise Transformation

1. contrast streching.
2. Gray level slicing
3. Bit plane slicing.

Contrast stretching

→ to increas dynamic range of the gray level

→ also known as normalization. The quality of image in enhanced by stretching the range of intensity values.

input intensity

$$I_{new} = (I - Min) \frac{New\,max - New\,min}{max - min} + New\,Min.$$

→ The bright pixels become brighter, the dark pixels become darker.

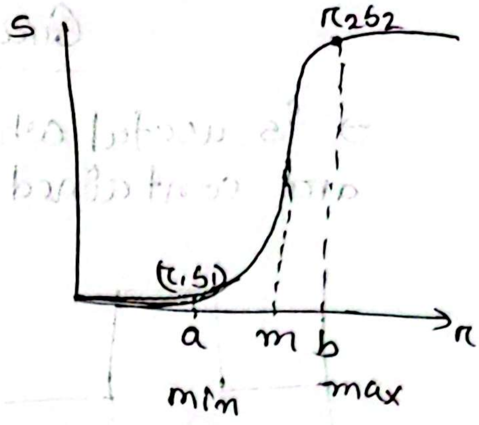
$(r_1, s_1) = (r_{min}, 0) = (a, 0)$

$(r_2, s_2) = (r_{max}, L-1) = (b, L-1)$

$r_{min} = a$. m = Contrast strechpoint

$r_{max} = b$

$r_1 < r_2$ $s_1 < s_2$.



* Input → 8bit image have range [80 – 150]

After contrast stretching → [0 – 255]

Lets I = ~~80~~ 100.

$$I_{new} = (I - min) \frac{NewMax - NewMin}{Max - min} + New\ min.$$

$$= (100 - 80) \frac{255 - 0}{150 - 80} + 0.$$

$$= 20 \times \frac{255}{70}$$

$$= 72.8$$

$$\approx 73.$$

* Suppose we have following intensity $a = 90$ $b = 180$, $m = 100$

Ans:

① If $r > 180$ → intensity will be 255

② if $r < 90$ → " " be 0

③ if r is between $a = 90$ & $b = 180$, T applies as follow.

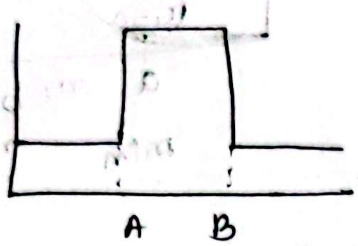
$r < 100$ → closes to zero (darker)

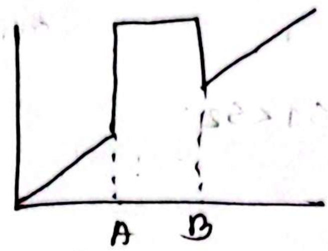
$r > 100$ → closes to 255 (brighter).

$\boxed{0.09 + 0.02}$ (0.09)

Gray/Intensity level slicing

→ is useful when different features of an image are contained in different Gray levels.





Chapter-3

Smoothing → Removes fine details.

What is kernel/ mask:

→ Represented by 2-D matrix.

→ mask usually of the order of 1x1, 3x3, 5x5, 7x7

→ Always in odd number.

→ known as filter/ mask/ window / kernel / box.

There are two types of filtering

1. Smoothing Spatial Filtering [loss pass]
2. Sharpening Spatial Filtering [High pass]

Why filters are used?

1. for blurring and noise reduction.
2. for edge detection and sharpness.
3. Enhance image
4. Extract information from image → Texture, edge, distin
5. Detect patterns → Template matching.

Average /mean Filter (linear)

→ also know as box filter.

→ has following properties

* It must be odd ordered
* The sum of all element should be 1.
* All element should be same.

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

3x3

To get border pixel:

① Zero padding.

② Border padding / Pixel replication.

③ Dicard border pixel.

What does mean / Box filter do

* Replaces each pixel with an average of its neighborhood
* Achieve smoothing effect (remove sharp features)
* Adds a 'softer' look to an image.

Weighted Smoothing Filters

→ different weights in the averaging function.

→ 4-neighbor mask will be half of the center weight and ~~1/4 of~~ diagonals will be 1/4 of the weight.

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussian Filter

→ Gaussian smoothing operator perform a weighted average of sosurrounding pixel based on Gaussian distribution.

→ remove Gaussian noise.

→ reduce sharpness; better than average filter.

→ kernels 3X3 5X5, 7,7.

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \qquad \frac{1}{273}\begin{bmatrix}1 & 4 & 7 & 4 & 1\\ 4 & 16 & 26 & 16 & 4\\ 7 & 26 & 41 & 26 & 7\\ 4 & 16 & 26 & 16 & 4\\ 1 & 4 & 7 & 4 & 1\end{bmatrix}$$

Order Statistic Non linear Filtering

→ min
→ max
→ median
- → No new pixel value introduced
- → Never replace with largest or smallest value
- → Removes spikes: Good for impulse, salt and pepper noise
- → Non linear filter
- → Blurring + Reduce noise

Common types of noise

① salt and pepper → White and Black dot / pixel
② Impulse noise → random occurances of white pixel
③ Gaussian noise → variations in intensity drawn from a Gaussian normal distribution

Sharpening spatial Filters.

→ Highlight fine details.

→ Remove blurring from image.

→ Highlights edges.

→ Useful for emphasizing transitions in image intensity.

Applications

① Photo enhancement.

② medical image visualization.

③ Industrial defect detection.

④ Electronic printing

⑤

(*) The 2nd derivative is more useful for image enhancement than the 1st derivative → stronger response to fine detail

Laplasian ⟶ 2nd derivate

SoBel → use of 1st derivative.

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

→ $g(x,y) = f(x,y) - \nabla^2 f$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

SOBEL

→ provides differencing and smoothing effect of an image.

→ 3x3 convolution kernels.

$$\frac{\partial f}{\partial y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \frac{\partial f}{\partial x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

G_y — Extract horizontal edges

G_x — Extract vertical edges.

$$G = |G_x| + |G_y|$$

→ Typically used for edge detection.

Difference Filter

→ Also called as Emboss filters

→ Enhances the details in the direction specific on the mask selected.

→

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

Vertical — Horizontal — Diagonal — Diagonal

Histogram

→ The histogram function is defined over all possible intensity level

→ For each intensity level, its value is equal to the number of the pixels with that intensity.

Applications:

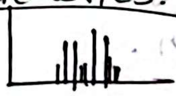
→ Image Enhancement.
→ Compression
→ segmentation.
→ tool for real time image processing.

Color image histogram 2 types

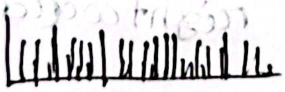
① Intensity histogram :
  → convert color image to gray scale.
  → Display histogram of gray scale.

② Individual color channel Histogram.
  → 3 Histogram (R, G, B)

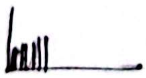
4 basic intensity characteristics:

Low Contrast images 

Histogram is narrow and centered toward and middle of the gray scale.

High contrast : 

→ Covers broad range of the gray scale and the distribution of pixels is not too far from uniform with very few vertical line being much higher than others.

Dark Image:  concentrated on the low side

Bright image:  concentrated on the high side

Why Important?

→ fast way to analyze the overall image quality.

→ By processing the histogram an image can be quickly and automatically improved.

Use of histogram

① whether image was scanned properly or not.

② idea about tonal distribution of an image.

③ equalization is applied to improve the appearance

④ tells us about object.

⑤ helps us to select threshold value for object detection

⑥ used for image segmentation.

Histogram Processing Techniques

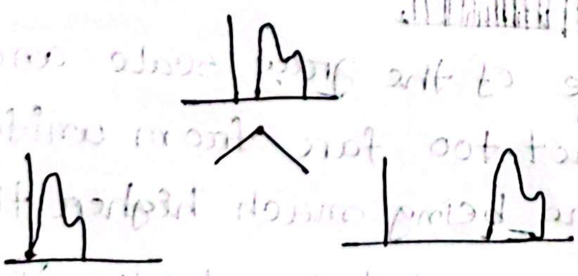
① Histogram sliding

② Histogram stretching

③ " Equalization.

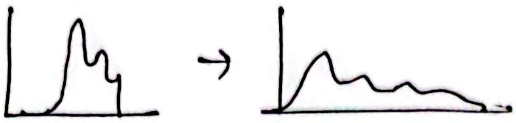
Sliding

Shifted toward rightwards or left word.

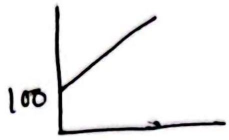


Stretching

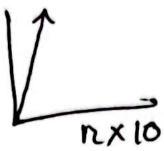
increasing contrast.



$T(r) = r + 100$ →



$T(r) = r \times 100$ →



Equalization → why important ?

Chapter-8 ~~Ari~~ Compression

(*) Arithmatic coding is a lossless compression technique that encodes data by creating a code string which represents a fractional value on the number line betn 0 and 1.

→ Non block coding:

→ An entire sequence symbols (string of symbol) is mapped to a single arithmatic number (code word)

→ This coding can achieve theoretically higher compression rates than Huffman codes.

→ Variable length code

→ Error-free Compression technique.

Data compression

→ refers to the process of reducing the amount of data required to represent a given quantity of information.

Application of image compressions

1. Internet
2. Business
3. multimedia,
4. satellite imaging
5. medical imaging

by compressed

For - 15,

30 frame × (720 × 480) × 3 bytes/pixel = 31 MB

31104000 bytes/sec × 60^2 sec × 2 hours.

= 2204 GB.

27 → 8.5 GB dual-layer DVDs (12 cm disk) are needed to store it.

Data Redundancy:

The relative redundancy, $R_D = 1 - \frac{1}{C_R}$ C_R = Compression Ratio

→ if $n_1 = n_2$ $C_R = 1$, $R_D = 0$ → no redundancy.

→ if $n_1 >> n_2$ $C_R \to \infty$ $R_D = 1$ → high "

→ if $n_1 << n_2$ $C_R \to 0$ $R_D = \infty$ → underiable.

*** A compression ratio 10 (10:1) means, the first data set has 10 information carrying units (bits) for every 1 unit in the second (compressed) data set.

OR 1 unit → 10 bit info → compressed →

Spatial

Original bit required (256×256×8)

In Run length (256+256)×8

C = 128:1.

CS CamScanner

1. Coding redundancy → does not need all 8 bits.

2. Spatial redundancy → Information is unnecessarily replicated.

3. Irrelevant info → Information is not useful.

Information theory

A random event E with probability P(E) carries I(E) units of information,

$$I(E) = \log \frac{1}{P(E)} = -\log P(E).$$

Higher the uncertainty, higher the information content.

If we have a source of random events from a discrete set of events $(a_1, a_2, a_3 \dots a_j)$ with probabilities $P(a_1), P(a_2), P(a_3) \dots P(a_j)$ then average information per event or the entropy of the source,

$$H = \sum_{j=1}^{j} -P(a_j) \log_2 P(a_j).$$

Considering pixel intensity as random events.

$$H = \sum_{k=1}^{L-1} -P(r_k) \log_2 P_r(r_k)$$

L = Gray levels / # of intensity

r_k = input image intensity

$P_r(r_k)$ = is normalized histogram of input image

Entropy is the measurement of the average / at least information in an image required.

Variable length coding → Huffman
→ Golomb
→ Arithmetic

LZW coding → spatial

Bit plane coding → Run length coding.

Huffman coding Properties

1. Codes one symbol at a time.
2. Block code.
3. Instantaneously Decodable.
4. Uniquely Decodable.

Golomb coding $G_m(n)$

1. Used to represent non-negative integer input.
2. with exponentially decaying probability distributions
3. Can be optimally encoded.
4. Using a family of code.
5. Computationally simpler than Huffman Coding.

$$q = \left\lfloor \frac{n}{m} \right\rfloor$$

$$k = \lceil \log_2 m \rceil$$

$$c = 2^k - m$$

$$r = n \bmod m$$

Lempel-Ziv-Welch (LZW)

1. Error free compression technique.
2. Removes spatial redundancy.
3. Assign fixed-length code word to variable length sequence of source symbol.
4. It does not require any knowledge of probability of occurence.
5. LZW coding is used in the GIF, TIFF, PDF formats.

Run-length Encoding

1. Spatial Redundancy eliminated.
2. R two parts (start of new intensity, # of consecutive pixels having that intensity)
3. Small runs results in expansion instead of compression.

Run-length in BMP

→ Uses combination of encoded + absolute mode.
→ Either mode can appear anywhere in the image.
→ Encoded mode → 2 byte RLC → 1 byte run length, 2 byte – gray/color index
→ Absolute → 2 byte RLC → First byte → 0 second byte ↓

Second byte value	
0	End of line.
1	End of image
2	move to a new position.
3-255	specify pixels individually

Fidelity Criteria

→ refers to the standards or measures used to access how accurately an image or signal is reproduced or represented after processing, like compression or transmission.

These criteria are important in areas like image processing, audio processing and other fields where maintaining the quality of original data is crucial, especially after compression or manipulation.

2 types:

① objective fidelity criteia → Mathematically defined

② subjective fidelity criteria → based on human observers

$$f(x,y) \longrightarrow \text{compress} \longrightarrow g(x,y) \longrightarrow \text{Decompress} \rightarrow \hat{f}(x,y)$$

$$\hat{f}(x,y) = f(x,y) + e(x,y)$$

How close $\hat{f}(x,y)$ to $f(x,y)$?

Objective fidelity criteria

→ are quantitatives measures that evaluate the difference between the original and the processed image using mathematical function/formulas.

These critarica provide numerical values to describe image quality, often without human interpretations.

Subjective

→ rely on human visual perception to evaluate the quality of an image.

These criteria are based on how an observer perceives the similarity or difference between the original and processed image.

Calculation (Objective)

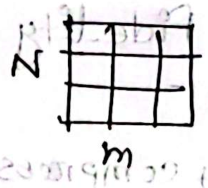
Error, $e(x,y) = \hat{f}(x,y) - f(x,y)$

$\hat{f}(x,y)$ = processed image

$f(x,y)$ = original image

Total Error betn two image.

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x-y) - f(x-y)\right]$$



Root mean Square Error:

$$e_{rms} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x-y) - f(x-y)\right]^2}$$

(*) Smaller value of erms is better compression

Signal to noise ratio:

$$SNR_{rms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y)\right]^2}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 6 & 5 \\ 8 & 9 & 10 \end{bmatrix}$$
$\hat{f}(x,y)$

$$\begin{bmatrix} 0 & 1 & 5 \\ 3 & 6 & 2 \\ 4 & 8 & 10 \end{bmatrix}$$
$f(x,y)$

* A larger SNR $\rightarrow$ better image.

$$\hat{f}(x,y)^2 = 1^2 + 2^2 + 3^2 + 4^2 + 6^2 + 5^2 + 8^2 + 9^2 + 10^2$$

$$\left[\hat{f}(x,y) - f(x,y)\right]^2 = (1-0)^2 + (2-1)^2 + (3-5)^2 + (4-3)^2 + (6-6)^2 + (5-2)^2 + (8-4)^2 + (9-8)^2 + (10-10)^2.$$

Chapter: 6 Colour Image Processing

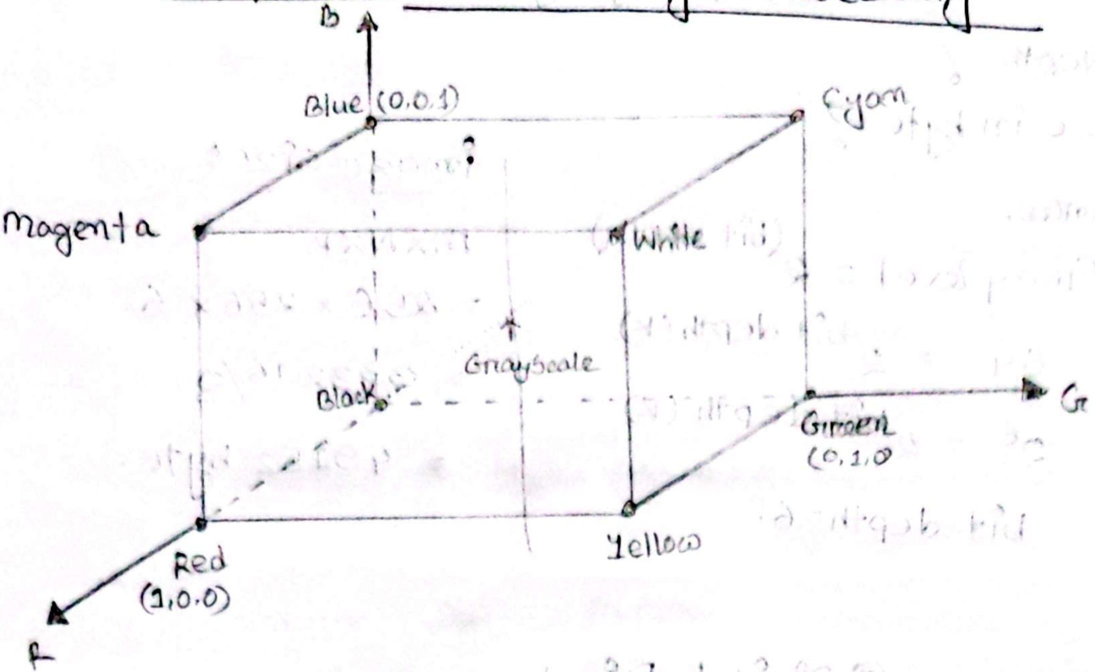
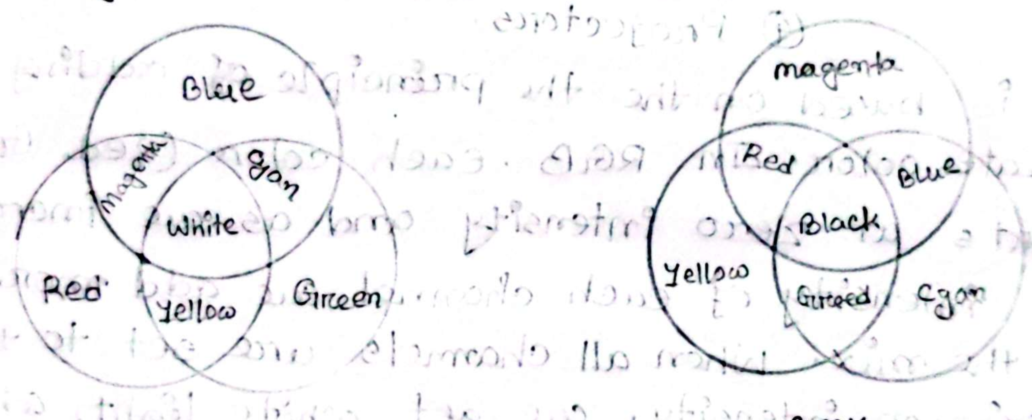
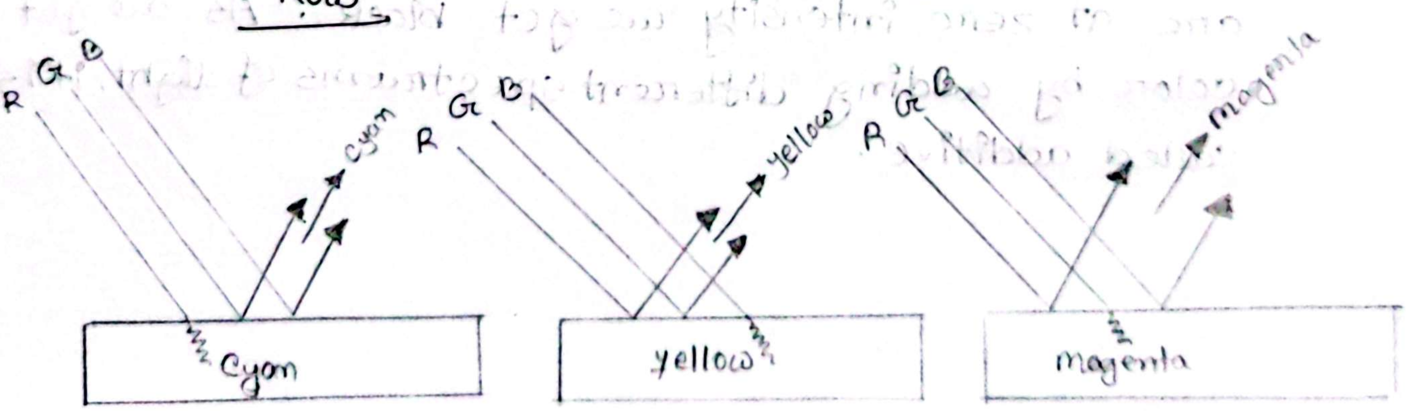


Figure: RGB Color Model. (Based on Cartesian Coordinate system)

Light intensity for human eye = (0·299 × Red) + (0·587 × Green) + (0·144 × Blue)



RGB CMY



Given, 256x256 image gray level 64.

Bit depth =?

size in byte =?

We know,

$$\text{Gray level} = 2^{(\text{bit depth})}$$

$$\Rightarrow 64 = 2^{\text{bit depth (K)}}$$

$$\Rightarrow 2^6 = 2^{\text{bit depth (K)}}$$

$$\Rightarrow \text{bit depth} = 6.$$

image size:

$M \times N \times K$

$= 256 \times 256 \times 6$

$= 393216/8$

$= 49152$ byte.

RGB used $\rightarrow$ ① Digital Display

② Computer monitors

③ Television Screen.

④ Projectors.

It is based on the the principle of adding light to create color. In RGB, each color (Red, Green, Blue) starts at zero intensity and as we increase the intensity of each channel we add more light to the mix. When all channels are set to their maximum intensity, we get white light, when all are at zero intensity we get black. As we get color by adding different spectrums of light, it is called additive.

Feature	Additive color Model	Subtractive
Primary color	Red, Green, Blue (RGB)	Cyan, magenta, Yellow (CMY).
Color Formation	Color formed by adding light	Colors are formed by absorbing light.
White creation	mixing all 3 colors (R+G+B)	Absence of pigment (reflecting all light)
Black creation	Absence of light (no RGB)	Combining all pigments (CMY)
Used In	Screens, digital display, lighting.	Printing, painting, color filter.

Chapter-6

out of all Cones of our eyes.

65% → Red light.
33% → Green Light. } sensitive.
2% → Blue light ——→ most sensitive

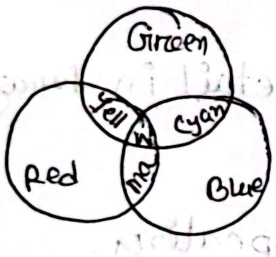
Additive

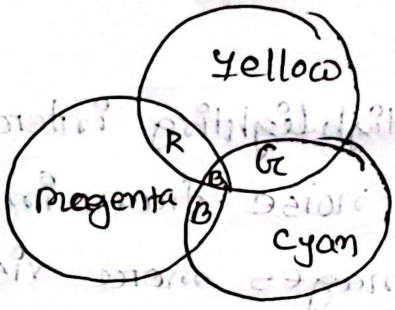
Combination of → red + green + blue → White.

Subtractive:

*Primary color → Cyan, magenta yellow.

* Black → cyan + magenta + yellow.





Red + Green = Yellow
Green + Blue = Cyan.
Blue + Red = Magenta

Cayan = White − Red.
Megenta = White − Green.
Yellow = White − Blue.

Erosion

$$A \ominus B = \{ z \mid (B)_z \subseteq A \}$$

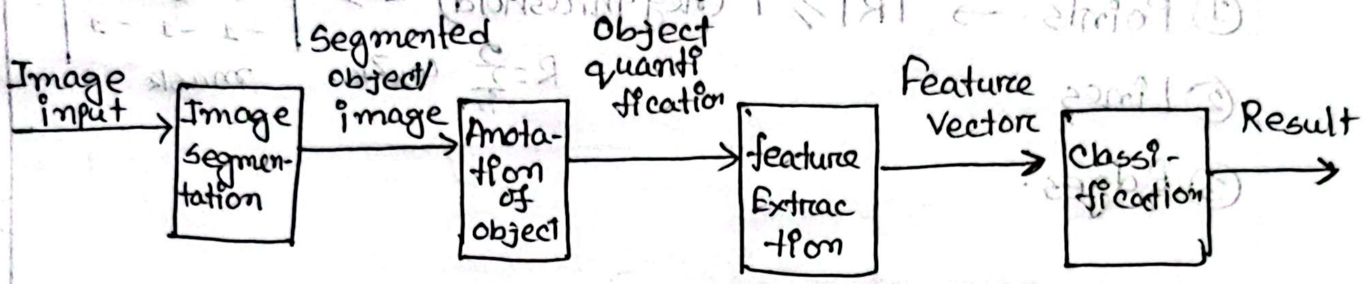
$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}$$

What is Image Segmentation

→ is an aspect of image processing

→ is a computer vision process.

→ First step in image analysis.



A typical image analysis pipeline.

→ Purpose → is to partition an image into meaningful region with respect to a particular application.

→ count

→ measure

→ study propertice. (intensity, texture)

Principle Approach of segmentation

1. Similarity →
   - → Thresholding → based on pixel intensity
   - → Region based → grouping similar pixel
   - → match based → comparison to a given templat

TRM

② Discontinuty → Partition an image based on sharp changes in intensity (such as edges in an image)

→ Edge based → Detection of edges that seperates regions from each orther

→ Watershed → find region corresponding to local minima in intensity.

3 types of Gray level discontineuities.

① Points. → $|R| \geq T$ (neg threshold) → $\begin{array}{ccc} \omega_1 & \omega_2 & \omega_3 \end{array}$ $\begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}$

$R = \sum_{i=1}^{9} \omega_k z_k$ mask

② Lines

③ Edges.

Line Detection

Horizontal

-1	-1	-1
2	2	2
-1	-1	-1

Vertical

-1	2	-1
-1	2	-1
-1	2	-1

diagonal (45°)

-1	-1	2
-1	2	-1
2	-1	-1

(-45°)

2	-1	-1
-1	2	-1
-1	-1	2

$|R_i| > |R_j|$ for all $j \neq i$ → line detect direction of mask i

Edge

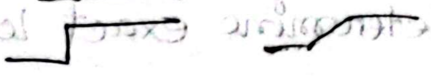
→ Abrupt change in the intensity of pixel

→ boundary of two region.

→ Discontinuity in image brightness or contrast.

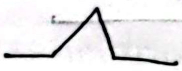
Edge is the boundary ~~of~~ between two homogeneous region

Edge Type (4)

① → Step Edge 

② → Ramp " 

③ → Ridge " 

④ → Roof " 

1st derivative → where an edge is

2nd " → show edge direction

Edge Detection

identify

— sudden change in image intensity.

Usage

→ Reduce unnecessary info of image while preserve structure of img

→ Extract important feature → Corners

→ Lines

→ Curves

→ Recognize objects, boundary. segmentation

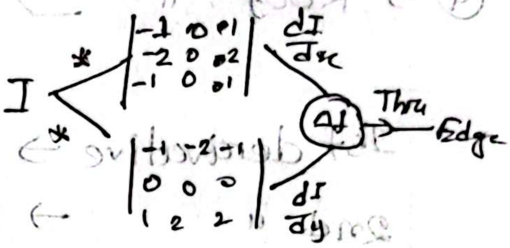
→ Part of computer vision and recognition.

3 main steps of Edge detection

1. Filtering (smoothing) → remove noise.
2. Differentiation (Edge sharpening) → where intensity changes i.e detects discontinuity.
3. Detection (Thresholding) → take decision . .
4. Localization → determine exact location of edge.

Only Gradient Based Edge Detection

→ simple to implement
→ detect edge and their direction.



→ sensitive to noise.
→ Not accurate in locating edges.

→ Too much details

Laplacian of Gaussian LoG

→ cheaper to implement → combine two filter
→ Doesn't provide info about direction of the edge.
→ Probability of false and missing edges.
→ Localization is better than Gradient operators.

steps

→ Smoothing → Gaussian filter
→ Enhance Edge → laplacian operator
→ zero crossing's denotes edge location
use → Linear interpolation approach to determine the sub-pixel location of the edge.

$$g(x,y) = \nabla^2 [G(x,y) * f(x,y)]$$

Canny Edge detection

3 main Criteria

① Good detection → Ability to ~~detect~~ locate and mark all real edges

② Good localisation → minimal distance betn real ~~e~~edge & detected edges

③ Clear response → Only one response per edge.

The Algo rithm → 5 steps.

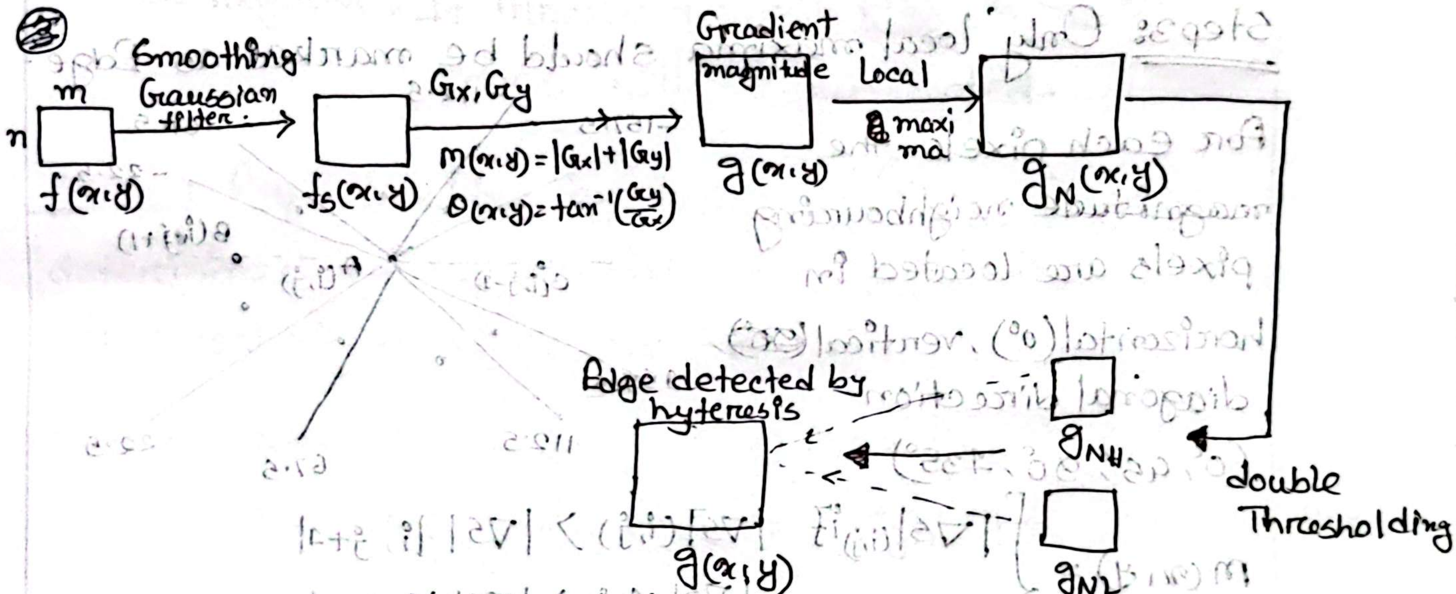
① Smoothing → remove ~~n~~noise

② Finding Gradients.

③ Non-maximum Suppression

④ Double Thresholding

⑤ Edge tracking by hysteresis.



Step1: Smoothing (Blurring of the edge image to remove noise).

Let $f(x,y)$ denote input image and $G(x,y)$ as Gaussian function: $G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$

Smoothed image $f_s(x,y) = G(x,y) * f(x,y)$.

Step 2: Finding Gradient (The edge should be marked where gradients of the image has large magnitudes)

Sobel:

horizontal

-1	-2	-1
0	0	0
1	2	1

Gy

vertical

-1	0	1
-2	0	2
-1	0	1

Gx

Prewitt:

-1	-1	-1
0	0	0
1	1	1

Gy

-1	0	1
-1	0	1
-1	0	1

Gx

Gradient magnitude:

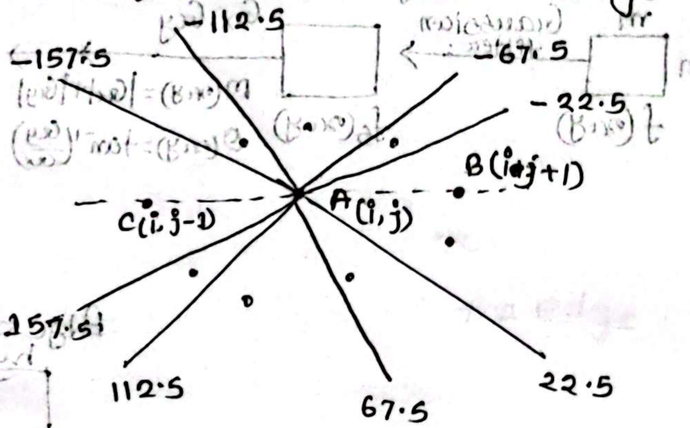
$$M(x,y) = \sqrt{(G_x)^2 + (G_y)^2}$$

or $|G_x| + |G_y|$

$$\alpha(x,y) = \tan^{-1}(G_y/G_x).$$

Step3: Only local maxima should be marked as Edge.

For each pixels the ~~magnitude~~ neighbouring pixels are located in horizontal(0°), vertical(90°), diagonal direction (0°, 45°, 90°, 135°)



$$M(x,y) = \begin{cases} |\nabla S|_{(i,j)} & \text{if } |\nabla S|(i,j) > |\nabla S|\,|i,j+1| \\ & |\nabla S|(i,j) > |\nabla S|\,|i,j-1| \\ 0 & \text{otherwise} \end{cases}$$

$g_N(x,y)$ will be the output.

Step 4: Double Thresholding (Potential Edges are determined by Thresholding)

To solve the problem of which edges are really edges and which are not canny uses Hysterisis Thresholding.

$$g_{NH}(x,y) = g_N(x,y) \geqslant T_H$$

$$g_{NL}(x,y) = g_N(x,y) \geqslant T_L$$

and $g_{NL}(x,y) = g_{NL}(x,y) - g_{NH}(x,y)$

The ratio of the high and low threshold should be two or three to one.

False positive: If threshold is too low than some unreal edges are detected as edges which are called fake positive edges.

False negative: If threshold is too high then actual valid edges will be eliminated.

Step 5: Edge tracking by hysteresis (Final Edges are determined by suppressing all edges that are not connected to a very strong edge.

For the edge pixel of $g_{NH}(x,y)$ and $g_{NL}(x,y)$

1. Any edge with it intensity > 'High' Threshold are the the 'sure edge'

2. Any edge with intensity < 'Low' threshold are the non-edge.

*3. The edges betaueen high and loco thereshold are classified as edges only if they are connected to the 'sure edge' otheraoise eleminated.

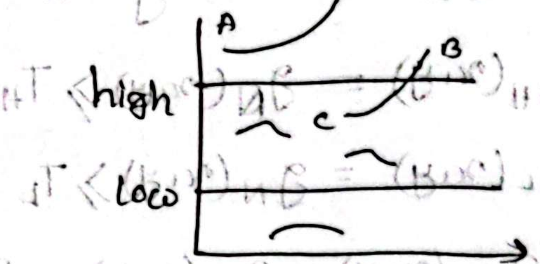
200	0	0
0	0	101
0	0	0

g_{NH}

0	0	57
0	45	0
50	0	0

g_{NL}

200	0	57
0	45	101
0	0	0



Edge Detection Algorithm:

1. Gradient Based → SOBEL
   → Perecuitt
   → Robert's Cross operator

2. Laplacian Based → Laplacian of Gaussian.

3. Hybrid → Gradient + Laplacian → Canny Edge detection

Line detection:

$$\begin{vmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{vmatrix} \quad \begin{vmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{vmatrix} \quad \begin{vmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{vmatrix}$$

horizontal   diagonal +45°

Point detection →

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

3 types of Gray level discontinuities:

① Points.

② Lines

③ Edges.