1) Discrete Cosine Transform.

⇒ The discrete Cosine transform (DCT) helps separate the image into parts of differing importance (with respect to the image's Visual quality). The DCT is similar to the discrete fourier transform: it transforms a signal or image from the spatial domain to the frequency domain.

DCT Encoding.

The general equation for a 1D (N data items) DCT is defined

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

where k varies between b/w 0 to (N-1)

Signal $x[n]$. which is denoted by $x_e[n]$ - so that the length of the extended sequence is 2N.

DFT of the extended sequence is $X_e[K] = \sum_{n=0}^{2N-1} x_e[n] e^{\frac{-j2\pi kn}{2N}}$

now we split the interval 0 to 2N-1 into 2

$$x_e[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{2N}} + \sum_{n=N}^{2N-1} x[2N-1-n] e^{\frac{-j2\pi kn}{2N}}$$

2) Discrete Fourier Transform -

FT of a sampled band~~width~~ limit function extending form $-\alpha$ to $+\alpha$ is a continous periodic function that also extends from $-\alpha \to +\alpha$

$$\tilde{F}(\mu) = \int_{-\alpha}^{+\alpha} f(t) e^{-j2\pi\mu t} dt$$

Substituting $f(t)$

$$\tilde{F}(\mu) = \int_{-\alpha}^{+\alpha} f(t) e^{2j2\pi\mu t} dt \qquad \int_{-\alpha}^{+\alpha} \delta(t)\, dt = 1$$

$$= \int_{-\alpha}^{+\alpha} \sum_{n=-\alpha}^{+\alpha} f(t)\, \delta(t - n\Delta T)\, e^{-j2\pi\mu t} dt.$$

$$= \sum_{n=-\alpha}^{+\alpha} f_n e^{-j2\pi\mu n\Delta T}.$$

Though $\tilde{F}(\mu)$ is a continuous, f_n is a discret. So $F(\mu)$ of 1 period and sampling of one period:

Let $\mu = 0$ to $\mu = \frac{1}{\Delta T}$ $\qquad \therefore \mu = \frac{m}{M\Delta t}$, $m = 0, 1, 2 \cdots M-1$

Substitution.

$$f_m = \sum_{n=0}^{M-1} f_n e^{-j2\pi mn/M}.$$

This is DFT. m & n are discrete variables.

Distance Measures:

four pixels p, q and z, with coordinates (x, y), (s, t) and (v, w) respectively, D is the distance function or metric if

i) $D(p,q) \geq 0$ $\quad (D(p,q) = 0$ iff $p = q)$;

ii) $D(p,q) = D(q,p)$ and

iii) $D(p,z) \leq D(p,q) + D(q,z)$.

a) The Euclidean Distance between p and q is —

$$D_e(p,q) = \left[(x-s)^2 + (y-t)^2\right]^{1/2}$$

for this distance measure, the pixels having a distance less than or equal to some values r from (x,y) are the points contained in a disk of radius r centered at (x,y)

b) City Block Distance or Manhattan Distance or D_4 distance between p and q is defined as

$$D_4(p,q) = |x-s| + |y-t|$$

In this case, the pixels having a distance less than or equal to r from (x,y) form a diamond centered at (x,y)

Eg. The pixels with D_4 distance ≤ 2 from (x,y) form the following pattern:

```
    2
  2 1 2
2 1 0 1 2
  2 1 2
    2
```

The pixels with $D_4 = 1$ are the 4 neighbours of (x,y).

c) The Chessboard Distance or Chebyshev Distance or D_8 distance between p and q is defined as —

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

In this case, the pixels with D_8 distance from (x,y) less than or equal to some value r form a square centered at (x,y).

For eg. the pixels with D_8 distance ≤ 2 from (x,y) (the center point) form the following contours of constant distance :—

$$\begin{matrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{matrix}$$

The pixels with $D_8 = 1$ are 8-neighbours of (x,y).

Define m-adjacency of 2 pixels. How does m-adjacency remove the ambiguities of 8-adjacency.

→ Two pixels p and q with values from V are m-adjacent if
i) q is in $N_4(p)$ or
ii) q is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

$V \rightarrow$ set of intensity values used to define adjacency
$V = \{1\}$.

Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used. For example, consider $\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$ for $V = \{1\}$. The three pixels at the top $\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$ shows multiple 8-adjacency i.e. ambiguous.

This ambiguity is removed by using m adjacency $\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$

What is image segmentation? it is applicable.

→ Segmentation subdivides an image into its constitute regions or objects. ~~The level to which~~

Let R represents the entire region occupied by an image. We may view image segmentation as a process that partitions R into n subregions $R_1, R_2 \ldots R_n$ such that —

a) $\bigcup_{i=1}^{n} R_i = R$.

b) R_i is a connected set, $i = 1, 2, \ldots n$.

c) $R_i \cap R_j = \emptyset$ for all i and j, $i \neq j$

d) $Q(R_i) = \text{TRUE}$ for $i = 1, 2, \ldots, n$

e) $Q(R_i \cup R_j) = \text{False}$ for any adjacent regions R_i and R_j.

The applications of segmentation are:—

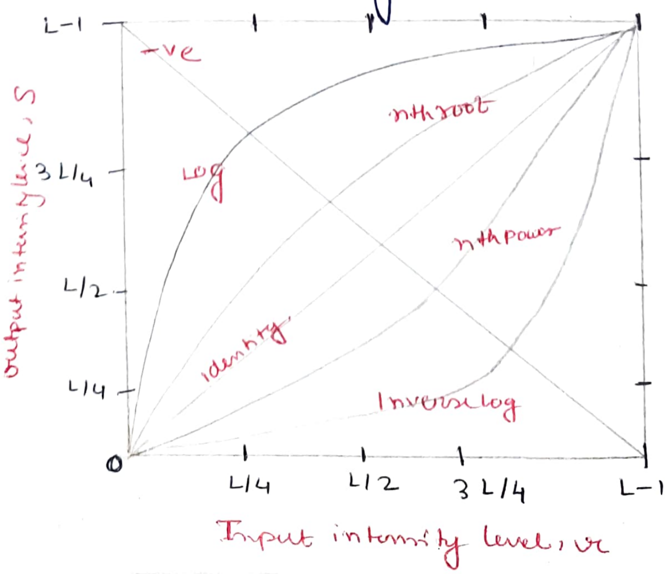
i) Detection of isolated points.

ii) Detection of lines and edges in an image.

Image Enhancement in Spatial Domain

1) Basic Grey Level / Intensity Transformation:

Intensity transformations are among the simplest of all image processing techniques. The values of pixels, before and after processing, will be denoted by r and s respectively.



There are 3 basic types of funtions used frequently for image enhancement:

i) Linear (negative and identity transforms)

ii) Logarithmic (log and inverse log transforms)

iii) Power-Law (n^{th} power and n^{th} root transformations).

The identity funtion is the trivial case in which output intensities are identical to input intensities.

The negative of an image with intensity levels in the range $[0, L-1]$ is obtained using image negative transformation

$$s = L - 1 - r$$

Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative.
This processing is suited for enhancing white or gray detailed embedded in dark regions of an image.

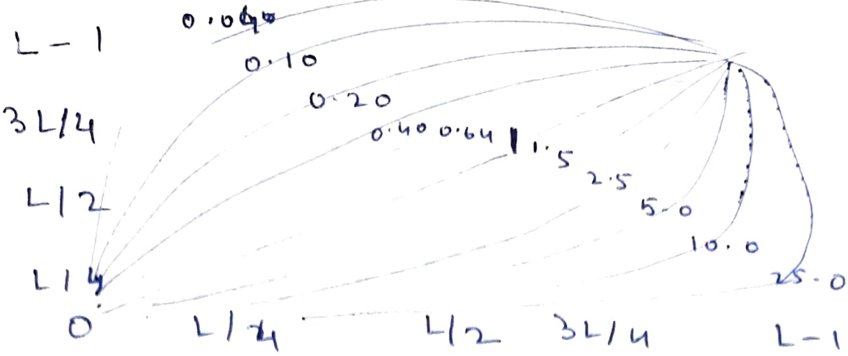
Log transformation's general form $s = c \log(1+r)$ where c is a constant, and it is assumed that $r \geq 0$.
The shape of the log curve that this transformation maps a narrow range of low intensity values in the input into a wider range of output levels.

This processing is used to expand the values of dark pixels in an image while compressing the higher-level values. The opposite is true of the inverse log transformation.

Power law (Gamma) transformations have the basic form

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

where c and γ are positive constants. Sometimes equation is also written as $s = c(r + \varepsilon)^{\gamma}$ to account for an offset.
Plots of s versus r for various values of γ are given below.
As in the case of the log transformation, power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.



The curves generated with values of $\gamma > 1$ have exactly the opposite effect as those generated with values of $\gamma < 1$. Finally γ reduces to the identity transformation when $c = \gamma = 1$.

Piecewise Linear Transformation functions

A complementary approach to the rest of 3 transformations is to use piecewise linear functions.

The principle advantage of piecewise linear functions over the types of other functions is the form of piecewise functions can be arbitrarily complex.

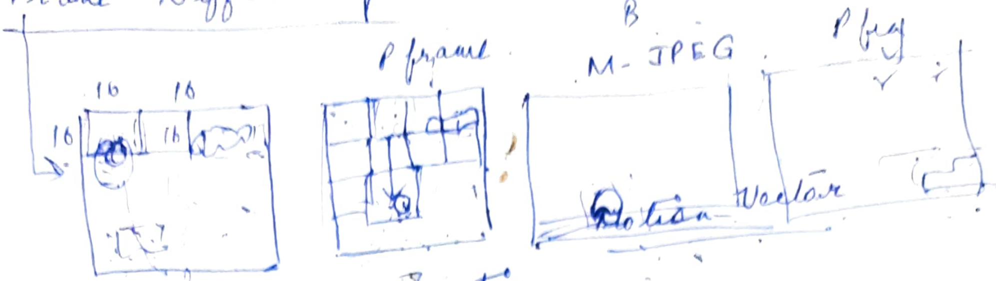
Disadvantage is that their specification requires considerably more user input.

Contrast stretching is one of the simplest piecewise linear functions. Low contrast images can result from poor illumination, lack of dynamic range in the imaging sensor or wrong setting of range lens, Contrast stretching is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

Intensity-level slicing. Highlighting a specific range of intensity in an image often is of interest. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in x-ray images. The process, often called intensity-level slicing, can be implemented in several ways, but most are variations of two basic themes. One approach is to display in one value (white) all the values in the range of interest and in another (black) all other intensities.

MPEG (video Compression Standard)

Frame Differencing



Motion Compensation

I frame, P frame, B frame
Intracoded
B frame → Bi-directional frame

I, Predictive frame

$\boxed{I}$ PPPBBBPP $\boxed{B}$ BBI

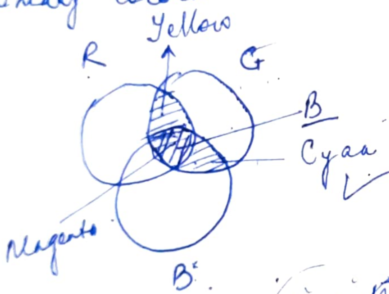
GOP → 12

Analog - NTSC - PAL - SeCAM

IPPBBBPPI

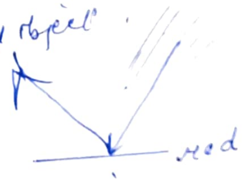
IPPPPBB

Primary Colour Model - RGB model



Colour gamut

Additive Colour Model



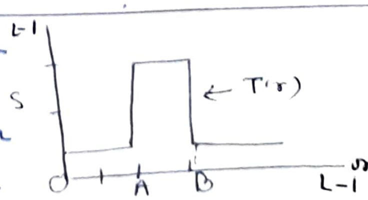
CMY $\boxed{K}$ -

YUV model

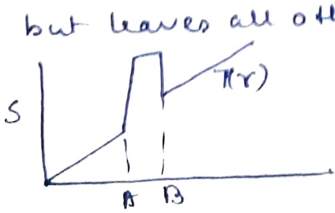
Y → luminance (brightness)

UV → chrominance (colour components)

This transformation produces binary image highlights [A, B] and reduces all other intensity to lower level.



The 2nd approach based on the transformation brightness (or darkens) the desired range of intensities but leaves all other intensities levels in the image is unchanged.



Bit-plane Slicing

Pixels are digital numbers composed of bits.
For ex. the intensity of each pixel in a 256 level gray scale image is composed of 8 bits.
Instead of highlighting intensity-level ranges we could highlight the contribution made to total image appearance by specific bits.
8 bit image is considered to have 8 one bit planes with plane 1 containing the lowest-order bit of all pixels in the images and plane 8 all the highest-order bits.

Histogram Processing.

The histogram of a digital image with intensity levels in the range $[0, L-1]$ is a discrete function $h(r_K) = n_K$, where r_K is the kth intensity value and n_K is the number of pixels in the image with intensity r_K.
A normalized histogram $\rightarrow P(r_K) = n_K / MN$, for $K = 0, 1, 2 \cdots L-1$,
$P(r_K)$ is an estimate of the probability of occurance of intensity level r_K in an image. The sum of all components of a normalized histogram is equal to 1.

There are 4 basic intensity characteristics; dark, light, low and high contrast.

In the dark image the components of the histogram are concentrated on the low side of the ~~lower~~ intensity scale.
Vice versa for light image.
An image with low contrast has narrow histogram located typically towards the middle of the intensity scale.
and vice versa for high contrast image.

Histogram Equalization is a technique for adjusting image intensities to enhance contrast.
Let f be image represented as a m_r by m_c matrix of integer pixel intensities ranging from 0 to $L-1$. L is the no. of possible intensity values often 256.

Let P denote the normalized histogram of f

$$P_n = \frac{\text{no. of pixels with intensity } n}{\text{total no. of pixel}} \qquad n = 0, 1 \dots L-1.$$

or, $$g_{ij} = (L-1) \sum_{n=0}^{f_{ij}} P_n$$

Correlation and convolution

Correlation is a process of moving a filter mask over the image and computing the sum of products at each location, exactly. The mechanism of convolution are the same, except that the filter is first rotated by 180°.

Correlation

f w

a) 00010000 12328

↓

b) 0 0010000 0
12328
↑ starting position allignment

Convolution

f w − 180°

00010000 82321

00010000
82321

f → function w → filter.

b shows the starting position here the parts of function donot overlap. The solution is to Pad f with enough 0's on either side to allow each pixel in w to visit every pixel in f.
if filter size is m, we need m−1 0's on either side of f.

$\overbrace{0000}^{\text{0 padding}}00010000\overbrace{0000}^{\text{0 padding}}$

c) 000000010000 0000
12328

0000 0001 0000 0000
82321

d) 0 000000100000000
12328
Position after 1 shift

0000 000 100000000
82321

e) 000000010000000 0
12328
final.

00 00000 10000000 0
82321

Full correlation result

000 82321 0000
↓
08232100

Full convolution

000123280000
↓
01232800

Note that it took 12 values of x ($x = 0, 1, 2 \ldots 11$) to fully slide w past f so that each pixel in w visited every pixel in f.

Correlation is a function of displacement of the filter.
Correlating a filter w with a function that contains all 0s ~~functions the~~ and a single 1 yields a copy of w but 180° rotated.

For 2D

$f(x, y)$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$w(x, y)$

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

Padding f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

initial position for w

$$\begin{matrix} 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 8 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

final for correlation

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 8 & 7 & 0 \\ 0 & 6 & 5 & 4 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

For Convolution

$$w(x, y) \quad \begin{matrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{matrix}$$

final result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

There are 2 smoothing filters. This are used for blurring and for noise reduction.

The output of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighbourhood of the filter mask. This filters are called averaging filter.

3x3

$$\frac{1}{9} \times \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

4x4

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

The idea of here is that it is computationally more efficient to have coefficients valued 1. At the end of filtering the entire image is divided by 9.

An $m \times n$ mask would have a normalizing constant equal to $1/mn$

A spatial averaging filter in which all coefficient are equal sometimes called box filter.

4x4 mask yields a weighted average.

General implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ is

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

$x = 0, 1, 2 \ldots M-1 \quad y = 0, 1, 2 \ldots N-1$

Four sharpening spatial filter.

First Derivative

1) must be 0 in areas of constant intensity

2) must be nonzero at onset of an intensity step or ramp

3) must be nonzero along ramp.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

*2nd. Derivative

1) must be zero in constant areas

2) must be nonzero at onset and end of intensity step.

3) must be zero along ramp

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial f'(x)}{\partial (x)} = f'(x+1) - f'(x)$$

$$= f(x+2) - f(x+1) - f(x+1) + f(x)$$

$$= f(x+2) - 2f(x+1) + f(x)$$

Substract 1 $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$.

Conclusion on applying 1st order & 2nd derivative for image segmentation

1) First order derivatives generally produce thicker edges.

2) 2nd order derivative have a strong response for thin line, point & noise.

3) 2nd order derivatives produce a double-edge response at ramp and step in intensity

4) The sign of the 2nd derivatives can be used to determine whether a transition is to an edge is from light to dark or vice versa.

Detection of Isolated points

Laplacian

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

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

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

Using Laplacian mask

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

we can detect point at location (x,y) on which the mask is centered if the absolute value of the response of the mask at that point exceeds a specific threshold.

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise.} \end{cases}$$

Line Detection

We use the same mask.

Laplacian image contains a negative values, scaling is always necessary for display

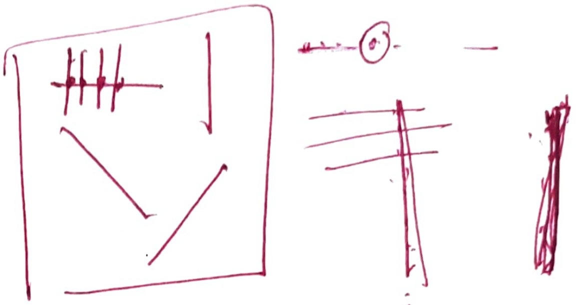
Line detection masks

$$\begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix} \qquad \begin{matrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{matrix} \qquad \begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix} \qquad \begin{matrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{matrix}$$

Horizontal $\quad$ $+45°$ $\quad$ vertical $\quad$ $-45°$

Basic edge detection techniques

$$\begin{matrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{matrix}$$



Roberts

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5)$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6).$$

Roberts operators are based on implementing the diagonal differe

$$\begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix} \qquad \begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix} \qquad 2\times 2$$

Prewitt

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

$$\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$$

Sobel

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7).$$

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \qquad \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

The Prewitt masks are simpler to implement but slight Computational differen between Sobel & prewitt.

Sobel masks have better noise suppression characteristics makes them preferable because noise suppression is an important issue when dealing with derivatives.

Modified prewitt and sobel. 12

$$\begin{matrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix} \qquad \begin{matrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{matrix} \qquad \begin{matrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{matrix}$$

They have their strongest responses along the diagonal direction.

Region based Segmentation

Region Growing is a procedure that groups pixel or subregions into larger regions based on predefined criteria for growth. The basic approach is to start with a set of seed points and from these grow regions by appending to each seed those neighboring pixels that have predefined properties similar to the seed.

Problem in region growing is the formulation of a stopping rule. Region growth should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as intensity values, texture, and color are local in nature and do not take into account the history of region growth. Additional criteria that increase the power of a region growing algorithm utilize the concept of size, likeliness between candidate pixel and pixels grown so far and the shape of the region being grown.

Basic region growing algo for 8-connectivity.

1. Find all connected components in $S(x,y)$ and erode each connected component to 1 pixel. Label them as 1 and others as 0

2. Form an image f_Q such that at a pair of coordinates (x,y) let $f_Q(x,y) = 1$ if the input image satisfy the given Q predicate else $f_Q(x,y) = 0$.

3. Let g be an image formed by appending to each seed point in S all the 1 valued points in f_Q that are 8-connected to that seed point.

4. Label each connected component in g with a different region label.

Region splitting & merging.

This is alternate to Region growing. This subdivides an image initially into set of arbitrary, disjoint regions and then merge and/or split regions in an attempt to satisfy the conditions of segmentation.

Let $R \rightarrow$ entire image

$Q \rightarrow$ selected predicate.

Divide R succesively into smaller and smaller quadrant regions so that, for any R_i, $Q(R_i)$ = True.

If $Q(R)$ = False, divide image into quadrants.

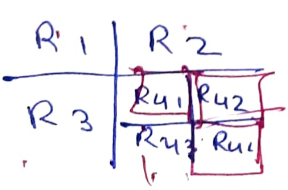
If Q is false for any quadrant, subdivide that.

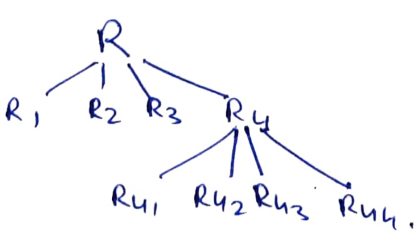
This is splitting technique and called quadtrees.

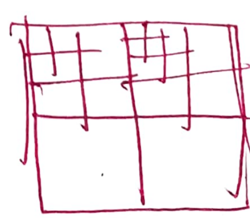
If only splitting is used, the final partition normally contains adjacent regions are identical properties. This drawback can be remedied by allowing merging as well as splitting.

Suppose some regions satisfy Q

That is, two adjacent region R_j & R_k are merged only if $Q(R_j \cup R_k)$ = True,







Basic relationship between Pixels. * *

A pixel p at Coordinate (x, y) has 4 horizontal and vertical neighbours whose coordinates are

$$(x+1, y)\ (x-1, y)\ (x, y+1)\ (x, y-1)$$

This set of pixel called 4-neighbour $N_4(P)$

4 diagonal neighbours

$$(x+1, y+1)\ (x+1, y-1), (x-1, y+1)\ (x-1, y-1)$$

These points together with 4 neighbour are called $N_8(P)$.

$N_8 = N_4(P) \cup N_D(P)$

Let v be the set of intensity values.

In binary image $V = \{1\}$.

Colour $V = \{0 - 255\}$

4-adjacency. Two pixels p and q with values from V are 4 adjacent if q is in the set $N_4(P)$

8-adjacency. Two pixels p and q with values from v are 8 adjacent if q is in $N_8(P)$.

$L = 16$ $M \times N = 256 \times 256 = 65536$.

r_k	n_k
$r_0 = 0$	5
$r_1 = 1$	4
2	12
3	

$P_r(r_k) = n_k / MN$.

r	0	1	2	3	4	5	6	7	8	9	10	11	12
$n(r_i)$	0	0	0	0	1	9	0	0	0	0	0	2	4
P(r) $n(r)/(4\times4)$	0	0	0	0	1/16 0.0625 0.25	9/16 0.5625 2.8125	0	0	0	0	0	2/16 0.125	4/16 0.25

cdf

$$\sum_{k=0}^{L-1} P(r_k).$$

$$\begin{bmatrix} 3 & 2 & 4 & 5 \\ 7 & 7 & 8 & 2 \\ 3 & 1 & 2 & 3 \\ 5 & 4 & 6 & 7 \end{bmatrix}$$ 1 to 8

Total no. of pixels associated.

1	2	3	4	5	6	7	8	9	10	Total
1	3	3	2	2	1	3	1	0	0	1
0.0625	0.1875	0.1875	0.125	0.125	0.625	0.1875	0.0625			#
0.0625	0.25	0.4375	0.5625	0.6875	0.75	0.9375	1	1		9 18 10

Probability.

no. of pix / 16

Calc Cumulative probability

image

5	4	12	5
5	5	12	5
5	12	12	11
5	5	11	5

Total gray pixels = 16   L - 1 = 15

Probality of occurance =

		$P(r_k)$		$C(r_k)$	$S=(L-1)C(r_k)$
0	0	0	0	0	= 0
1	0	0	0	0	= 0
2	0	0	0	0	= 0
3	0	0	0	0	= 0
4	1	1/16	0.0625	0.0625	= 0.9375
5	9	9/16	0.5625	0.6250	= 9.375
6	0	0	0	0.6250	= 9.375
7	0	0	0	0.6250	= 9.375
8	0	0	0	0.6250	= 9.375
9	0	0	0	0.6250	= 9.375
10	0	0	0	0.6250	= 9.375
11	2	2/16	0.125	0.75	= 11.25
12	4	4/16	0.25	1	= 15

Rounding the value of S

we get.

Pixel	S
4	1
5	9
11	11
12	15

New image

9	1	15	9
9	9	15	9
9	15	15	11
9	9	11	9

$L = 8$ $\quad L-1 = 7$

$64 \times 64 \therefore M \times N = 4096$.

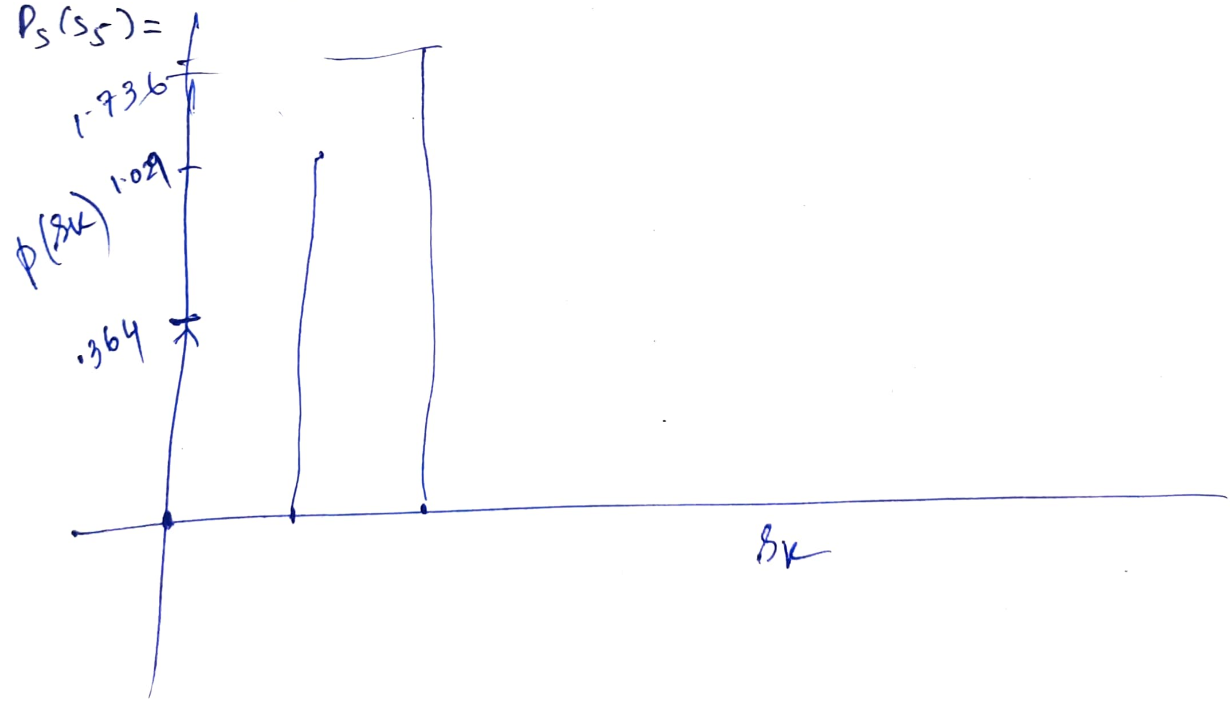
r_k	n_k	$P_r(r_k) = n_k/MN$	$(L-1)\sum P_r(r_j)$	rounded	
0	123	$123/4096 = 0.030$	0.21	0	S_0
1	78	$78/4096 = 0.019$	0.343	0	S_1
2	281	$= 0.068$	1.029	1	S_2
3	417	0.101	1.736	2	S_3
4	639	0.156	2.828	3	S_4
5	1054	0.257	4.627	5	S_5
6	816	0.199	6.02	6	S_6
7	688	0.167	7.189	7	S_7

~~$P_s(S_0) = 123 = P_s(S_1) = 0.030$~~

$P_s(S_0)$ (S_1) $= 123 + 78 = 0.049$.

$P_s(S_2) = 0.068$ $\quad P_s(S_3) = 0.101$ $\quad P_s(S_4) = 0.156$.

$P_s(S_5) =$



What is a Digital Image.

- a two dimensional function X and y are spatial coordinates

The amplitude of f is called intensity at point (x, y).

Digital Image processing -

Process digital images by means of computer, it covers low, mid and high level processes.

low — input and outputs are images

mid — outputs are attributes extracted from input.

high — an ensemble of recognition of individual objects.

Pixel — the elements of a digital image.

Fundamental steps of DIP.

Colour image processing

Compression

Morphological Processing → extracts image component.

Image restoration → improves appearance

Segmentation → Partitions image

Representation → Represents image for computer processing.

image enhancement

Object recognition

image acquisition

Spatial resolution — a measure of the smallest discernible detail in an image

Intensity resolution — The smallest discernible change in intensity level.

Interpolation — Process of using known data to estimate unknown values.

also known as resampling

A digital path from pixel p with coordinates (x_0, y_0) to pixel q with coordinat (x_n, y_n) is a sequence of distinct pixel with coordinates

$$(x_0, y_0)\ (x_1, y_1) \cdots (x_n, y_n).$$

where (x_i, y_i) & (x_{i-1}, y_{i-1}) are adjacent for $1 \leq i \leq n$.
n → length of path.

If $(x_0, y_0) = (x_n, y_n)$ → close path.

Two pixels p with (x_0, y_0) and q with (x_n, y_n) are said to be Connected in S if there exist a path.

where $\forall i,\ 0 \leq i \leq n,\ (x_i, y_i) \in S$.

What are the needs for image transform

→ Image transforms can be simple arithmatic operations which converts images from one representation to another.
like — Fourier, Hough, Histogram etc.

Digitization

A digitization process is often used to convert analog data, such as media, sound, image, and text into digital representation.
Digitization is done in 2 steps
Sampling and Quantization.
First step, data is sampled at regular intervals, such as the grid of pixels used to represent a digital image.
The frequency of sampling is referred to as resolution of the image. Sampling turns continuous data into discrete data.

second, each sample is quantified which determines no. of gray levels
The no. of quantized levels should be high enough for human perception of shading . .

Image Acquisition

Image Acquisition in image processing is defined as the action of retrieving an image from some source, usually a hardware, source so it can be passed through whatever process need to occur afterward.
Performing IA in image processing is always the first step in the workflow sequence because, without an image, no processing is possible.
One of the ultimate goal of this process is to have a a source of i/P that operates within such controlled and measured guidelines that the same image can, if necessary, be nearly perfectly reproduced under same condition.

image Smoothing (Spatial domain).

Smoothening is the blurring of image to reduce noise.
Blurring is used in preprocessing tasks, such as removal of small details for object extraction etc.
Noise reduction can be accomplished by blurring with a linear filter and non linear.

Smoothing linear filter.

The o/p of a smoothing, linear spatial is simply the average of the pixels contained in the neighbourhood of the filter mask. These are called averaging filter or lowpass filter.

Simply average all of the pixels in a neighborhood around a central value.

Especially useful in removing noise.

104	100	108
99	$\boxed{106}$	98
95	90	85

$\boxed{3 \times 3}$

$$\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$ filter.

$$e = 1/9 \times 106 + 1/9 \times 104 + 1/9 \times 100 + 1/9 \times 108 + 1/9 \times 99 + 1/9 \times 98 + 1/9 \times 95 + 1/9 \times 90 + 1/9 \times 85$$

$$= 98.3333.$$

This is repeated for every pixel in the original image to generate smoothed image.

Filter is used to remove noise.

median filter work better than averaging filter.

At the edges of an image we are missing pixels to form a neighborhood.

few approaches to deal with it

- omit those.
- Pad the image
- replicate border pixels
- Truncate the image

Median filter (nonlinear filter).

Based on ordering the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

Median filter are particularly effective in the presence of impulse noise

image Smoothening (frequency domain)

1) Ideal Lowpass filter

It passes all frequencies within a circle of radius DO from the origin and cuts off all other frequencies

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

where D_0 = +ve const.

$D(u,v)$ = Distance between center to point of frequency domain.

$$D(u,v) = \left[(u-P/2)^2 + (v-Q/2)^2\right]^{1/2}$$

for ILPF, the transition is very sharp which cannot be realized electronically but can be simulated electronically.

2) Butterworth Lowpass filter

The transfer function of a Butterworth lowpass of order n and with cut off D_0 is

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

Unlike ILPF, BLPF transfer function does not have a sharp discontinuity that gives a clear cut off between passed and filtered frequency. For filters with smooth transfer function, defining a cutoff frequency location at $H(u,v)$ is down to certain fraction of its max value is customary.

3) Gaussian filter

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

σ is a measure of spread about the center.

$\sigma = D_0$

i) $H(u,v) = e^{-D^2(u,v)/2D_0^2}$

when $D(u,v) = D_0$ the GLPF is down to 0.607 of its max value.

~~Image Sharpening~~

2 widely used and popular metrices used in image restoration field.

i) Blurred Signal to noise Ratio (BSNR)

The degradation modeled by blur and additive noise is reffered by a metric called BSNR)

If $f(m,n)$ reprents $M \times N$ image and blurred by the point spread function $h(m,n) \rightarrow y(m,n)$ then

$$BSNR = 10\log\left[\frac{\frac{1}{MN}\sum\left[[y[m,n] - \bar{y}[m,n]\right]^2}{\sigma_n^2}\right]$$

additive nois variance

mean of $y[m,n]$

ii) Improvement in Signal-to-noise Ratio (ISNR)

ISNR is used to test the performance of image restoration algo.

If $f[m,n]$ and $y[m,n]$ reprent original image and degraded then,

$$ISNR = 10\log_{10}\left[\frac{\sum_{mn}\left[f[m,n] - y[m,n]\right]^2}{\sum_{mn}\left[f[m,n] - \hat{f}[m,n]\right]^2}\right]$$

$\hat{f}[m,n]$ is the restored image. This metric can be used for simulation purpose only because the original image is assumed to be available which is not practical.

Image Sharpening (Spatial)

The principal objective of sharpening is to highlight transition in intensity. Uses of image sharpening varyond include applications ranging from electronic printing and medical imaging to industrial inspection & autonomous guidance in military systems.

Implementation is done by digital differentiation

1st derivative

1) must be 0 in areas of constant

Image Sharpening (frequency domain).

High pass filters

High frequency component in an image → edges & other abrupt changes. It attenuates low frequency components. Without changing the high frequency components.

Filter function $H(u,v)$ → discrete fn of size $P \times Q$

i.e $u = 0, 1, 2, \ldots, P-1$ and $v = 0, 1, 2, \ldots, Q-1$

A high pass filter is obtained from a LPF as

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

H_{LP} → Transfer fn of LPF

When the LPF attenuate HPF passes and vice-versa

1. Ideal HPF

A 2-D ideal HPF is defined as

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

where, $D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2\right]^{1/2}$

& D_0 = cut off frequency.

Butterworth Highpass Filter

A 2-D Butterworth highpass filter (BHPF) of order n and cut off frequency D_0

$$H(u,v) = \frac{1}{1+[D_0/D(u,v)]^{2n}}.$$

where $D(u,v) = [(u-P/2)^2 + (v-Q/2)^2]^{1/2}$.

Gaussian filter

The transfer function of GHPF with cutoff locus at a distance D_0 from center

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}.$$

Need for image transforms

Image transforms can be simple arithmetic operation on images or complex operations which converts images from one representation to another.

Mathematical operation — arithmetic fourier, hough etc

Histogram modification, Image rotation etc.

Image transforms are used in preprocessing, Data compression, feature Extraction.

The aim of image transform is to represent the input image in the form of linear combination of a set of basis image

Properties of DFT

i) Symmetric property ; ii) fast transform Separability, Translation ; Computational capacity is $N \log_2 N$.

FT

- Requirements
i) Function must be Continuous & integrable.

DCT

DCT is a forward transformation

$$g(x,y,u,v) = \alpha(u) \cdot \alpha(v) \cdot \cos[(2x+1)u\pi/2N]$$
$$\cdot \cos[(2y+1)v\pi/2N]$$
$$= f(x,y,u,v)$$

inverse as forward transformation are identical they are Separable aswell.

Properties

i) It is real orthogonal.
$$C = C^* \Rightarrow C^{-1} = C^T$$

ii) Fast transf
iii) Good for energy compaction
ii) used for designing transform code & wiener filter
iv) JPEG Compression

Haar transform

$h_x(K)$ are defined on Continuous interval $x \in [0,1]$ and for $K = 0, 1, \cdots N-1$ where $N = 2^n$.

image restoration

The purpose of IR is to compensate for or undo defects which degraded an image. Degradation comes in many forms such as motion blur, noise etc.

methods to restoring image

i) inverse filter restoration is good when noise is not present

ii) Weiner filter provides the optimal trade-off between de-noising and inverse filtering.

Watershed image segmentation

The term watershed refers to a ridge that divides areas drained by different river system.

Watershed of a grayscale image is analogous to the notion of a catchment basin of a heightmap.

There are diff technical def.

In graph, watershed lines maybe defined on nodes.

Watershed may also be defined as continuous domain.

Watershed algorithm is used in image processing for image segmentation.

Advantage is that it always provides closed contours, which is very useful in image segmentation,

WT requires low computation times in comparison with other methods.

The principal objective is to find the watershed lines

Local & Global thresholding.

The segmented image $g(x,y)$ is given by

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases}$$

When T is constant applicable over an entire image, the process given is reffered as global thresholding.

When the value of T changes → variable thresholding

Local is used to denote variable in which the value of T at any point (x,y) in an image depend on properties of a neighbourhood of x,y.

if T depends on spatial coordinates (x, y) themselves then variable thresholding is call adaptive T

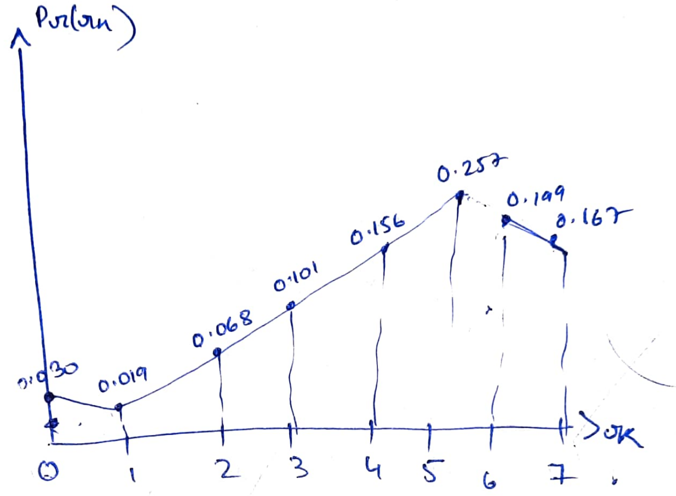
<u>2012-2013</u>

$L = 8$ $\quad L-1 = 8-1 = 7 \quad M \times N = 64 \times 64 = 4096.$

Range $= (0, 7)$

r_k	n_k	$P_r(r_k) = n_k / MN$
0	123	0.030
1	78	0.019
2	281	0.068
3	417	0.101
4	639	0.156
5	1054	0.257
6	816	0.199
7	688	0.1678

Histogram for original



Histogram equalization transformation function

$$S = T(r_k) = (L-1) \sum P_r(r_k).$$

$$S_0 = L-1 \sum_{k=0}^{0} P_r(r_k) = 7 \times 0.030 = 0.21$$

$$S_1 = L-1 \sum_{k=0}^{1} P_r(r_k) = 7 \times 0.019 + 7 \times 0.030 = 0.343$$

$$S_2 = L-1 \sum_{k=0}^{2} P_r(r_k) = 7 \times 0.068 + 7 \times 0.019 + 7 \times 0.030 = 0.819$$

$$S_3 = L-1 \sum_{k}^{3} P_r(r_k) = 7 \times 0.101 + 7 \times 0.068 + 7 \times 0.019 + 7 \times 0.030 = 1.526$$

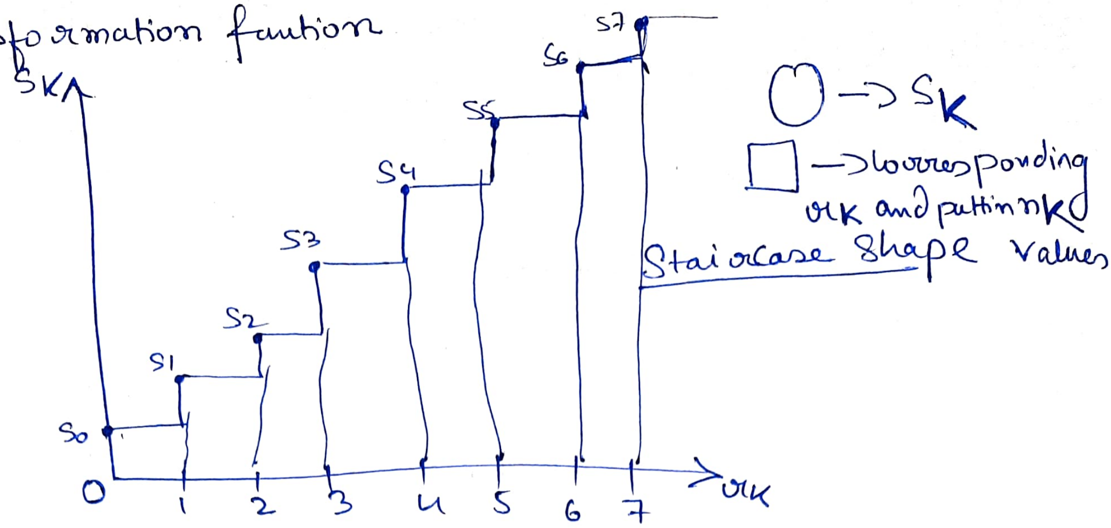
$$S_4 = L-1 \sum_{0}^{4} P_r(r_k) = 7 \times 0.156 + 7 \times 0.101 + 7 \times 0.068 + 7 \times 0.019 + 7 \times 0.030 = 2.618$$

$$S_5 = L-1 \sum_{0}^{5} P_r(r_k) = 7 \times 0.257 + 7 \times 0.156 + 7 \times 0.101 + 7 \times 0.068 + 7 \times 0.019 + 7 \times 0.030 = 4.417$$

$$S_6 = L-1 \sum_{0}^{6} P_r(r_k) = 7 \times 0.199 + 7 \times 0.257 + 7 \times 0.156 + 7 \times 0.101 + 7 \times 0.068 + 7 \times 0.019 + 7 \times 0.030 = 5.8$$

$$S_7 = 7 \times 0.167 + 7 \times 0.199 + 7 \times 0.257 + 7 \times 0.156 + 7 \times 0.101 + 7 \times 0.068 + 7 \times 0.019 + 7 \times 0.030 = 6.9$$

Transformation function



Rounding the fractional value.

$\boxed{S_0} = 0$ $\boxed{S_1} = 0$ $\boxed{S_2} = 1$ $\boxed{S_3} = 2$ $\boxed{S_4} = 3$

$\boxed{S_5} = 4$ $\boxed{S_6} = 6$ $\boxed{S_7} = 7$

For S_0 values of n_0 and n_1 are taken

$P_s(S_0) = \frac{123+78}{4096} = 0.05$

$P_s(S_1) = \frac{281}{4096} = 0.0686$

$P_s(S_2) = \frac{417}{4096} = 0.102$

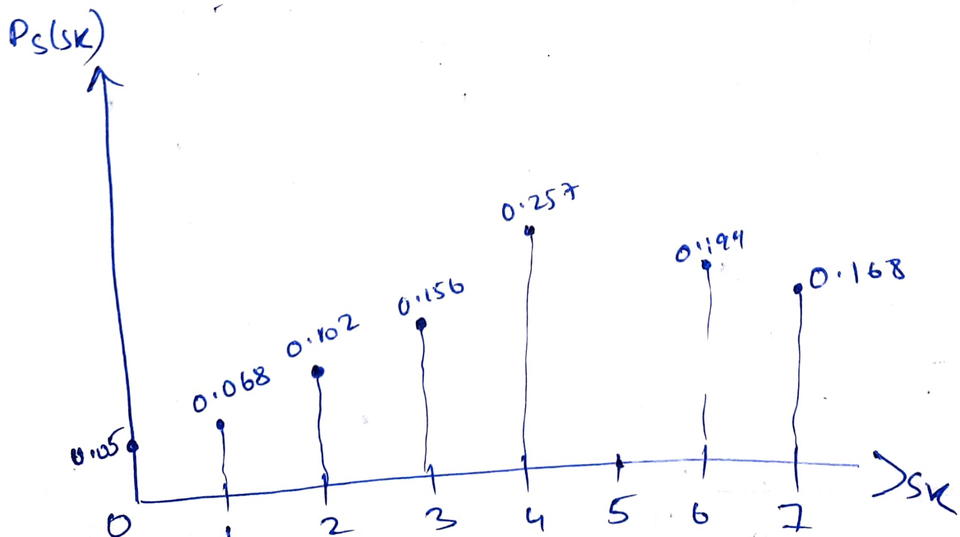
$P_s(S_3) = \frac{639}{4096} = 0.156$

$P_s(S_4) = \frac{1054}{4096} = 0.257$

$P_s(S_5) = \text{Null} = 0$

$P_s(S_6) = \frac{816}{4096} = 0.199$

$P_s(S_7) = \frac{688}{4096} = 0.168$



Matrix form

5	4	12	5
5	5	12	5
5	12	12	11
5	5	11	5

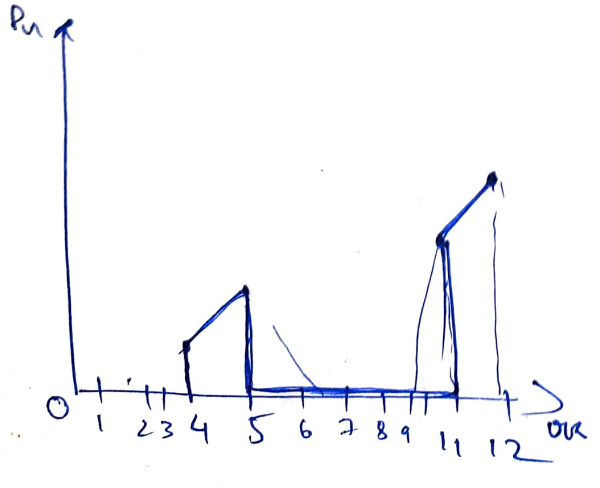
Total grey pixels = 16 = M×N

L-1 = 15

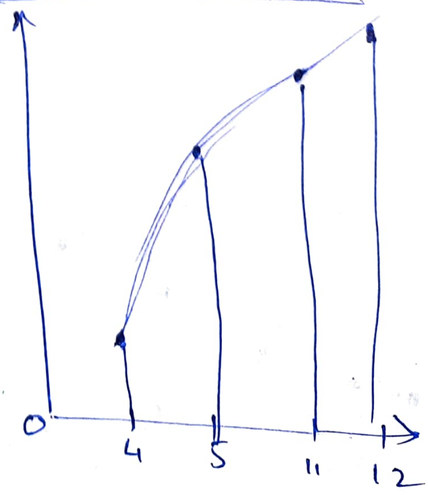
Probability of occurrence

r_k	n_k	$P(r_k)$	Cumulative frequency (r_k)	$S=(L-1)\,C_{r_k}$
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
$\boxed{4}$	1	1/16 = 0.0625	0.0625	0.9375
$\boxed{5}$	9	0.5625	0.625	9.375
6	0	0	0.625	"
7	0	0	"	"
8	0	0	.	.
9	0	0	.	"
10	0	0	0.625	"
$\boxed{11}$	2	0.125	0.75	11.25
$\boxed{12}$	4	0.25	1.00	15.00
13	0	0	"	"
14	0	0	"	"
15	0	0		

Histogram for P_{r_k} & r_k



Histogram for r_k & S.



Rounding the values

r_k	s
4	1
5	9
11	11
12	15

JPEG

Joint Photographic Experts Group standard for images baseline coding of photographic quality. Its lossy discrete cosine transform system uses quantized discrete cosine transform on 8x8 image & block, Huffman and runlength coding. Its one of the popular methods for compressing images.

JPEG Compression process is composed of the following steps:–

i) Block Preparation — An image is represented by one or more 2D arrays of pixel values. The block preparation step breaks each 2D array of the image into individual 8x8 pixels per block. This blocks are in preparation and next step is applied to each block.

ii) Source encoding (DCT & quantization).
2D DCT is used to convert each block from the spatial to frequency domain. First frequency component is called DC coefficient & to average of all pixelvalue, 2nd AC coefficient. Since 8x8 block of pixels occupy a small area variation will be small hence AC will be small or 0.

So far no info is lost. If image stored as such and then inverse transform is applied, the initial image will be recovered. This property is exploited in quantization phase by dropping higher spatial frequency coefficient in the transformed array.

JPEG standards include default quantization tables; it allows four customized tables to be use.
This step leads to data loss and reduces the fidelity of resulting imag

$$\text{DCT Coefficient} \Big/ \text{Quantization Coefficient} = \text{Quantized Coefficient}$$

iii) Entropy Encoding algo operate on a 1D string of values.
The o/p of quantization stage is however 2D.
Hence to apply entropy scheme image is converted to 1D vector known as vectoring –
upper left corner are non-zero lower right mostly zero.
Zig-Zag scan is row by row from ~~right~~ left to right. is performed.

Differential Pulse code modulation. only 1 DC per block.
DC is measure of the mean of 64 values of that block.
it is ∴ largest coefficient. Because of small area, DC coefficient varies slowly. To exploit this similarity, DC coefficient is encoded by DCPM mode.

Run-length Encoding After Quantization only some AC coefficient have survived other are 0. Those surviving values are to be stored. So RLE is applie

Huffman coding is the final step. This is applied to both encoded DC and AC coefficient within block.

Frame Packing block does ~~not~~ the final assembling of data and adds addition error checking codes before sending

MJPEG

Motion JPEG is a simple video compression scheme where each frame is coded separately as a JPEG image and then compined together into a video file. It uses only intra-frame compression where each frame is compressed using the lossy JPEG algo based on DCT.

Advantage is that its video quality is not degraded by fast or random motion in the video content as in MPEG based on inter frame conpression.

MPEG-1

Motion pictures Expert Group Standard for CD-ROM applications with non-interlaced video.

Frame prediction is based on the previous frame, next frame or an interpolation of both.

MPEG-2

for DVDs.

H.261

This was developed by ITU-T in 1993 for video telephony and conferencing applications in as ISDN environment.

H.261 is similar to MPEG-1 as it uses common Interface Format or Quater CIF and uses CCT based algo to remove intra frame redundary and motion compensation to remove interframe redundancy

H.262

This is MPEG-2 in 1994 was developed to provide video quality not lower than and upto HDTV quality.

Coding Structure is same as MPEG-1

Huffman Coding is the Variable length coding technique
It is used to construct a binary tree of the pixels with their occurance probability.
It is the technique of removing code redundancy
It assigns the shortest possible code word per source symbol.

MPEG

Motion Phot Picture Expert Group :- takes the advantage of temporal and Spatial redundancy.

I frame — Self contained.

Predictive (P) frame — requires previous I frame / P frame for encoding & decoding

Bidirectional (B) frame — requires both the info of I & P frames.

Histogram Equalization is a technique for adjusting pixel values or image intensities to enhance contrast

$$P_n = \frac{\text{no. of pixels intensity } n}{\text{total no. of pixel}}$$

Histogram of a digital image with gray level in the range [0, L-1] is a discrete function $h(r_k) = n_k$
A image histogram is a type that acts as a graphical representation of a tonal distribution of each pixel for its tonal value.

Median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the resultt of later processing.

MF preserves the edges while removing noise.

idea is to run through the signal entry by entry replacing each entry with the median of neighboring entry over signal.

lowpass filter averages out rapid changes in intensity.

Calculates pixel of all its 8 immidiate neighbour.

The result replaces the original value of the pixel.

∴ LPF looks a lot blurrer.

Blur is required cuz often images ca be noisy no matter how good the camera is.

Hough Transform is a feature extraction technique used in image analysis and DIP. The purpose is to find imperfect instances of objects within a class of shapes by a voting procedure.

Image Degradation is a process by which the original image is degraded is usually very complex.

The degradation is often modeled as a linear function which is often reffered as point spread function

i) Improper opening & closing of shutters

ii) Atmospheric of lens iii) Misfocus of lens iv) Relative motion between camera & object

Types of blur

i) Gauss blur — $h(x,y) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2+y^2}{2\sigma^2}}$

variance of blur.

This occurs due to long time atmospheric exposure.

ii) Out of focus blur — This is produced by a defocused optical system. It distributes a single point uniformly over a disk surrounding a point.

iii) Motion blur — It is due to relative motion between the recording devices & the scene. When an object or the camera is moved during light exposure, a motion blur is produced.

RGB (Red, Green, Blue): is an additive color model that is used for light emitting devices, such as CRT displays. RGB can be thought of as three grayscale images representing the light values of Red, Green and blue. Combining these three channels of light produces a wide range of visible colors.

YUV luminance - chrominance reduces the amount of information required to reproduce an acceptable video image. The principle advantage of using YUV for broadcast is that the amount of info needed to define a color television image is greatly reduced. This compression restricts the color ~~change~~ range in these images.

Y - luminance I - red - cyan channel Q magenta green channel.

YUV — U - blue V - red chan.

CMYK Cyan, Magenta, Yellow & black.

HSB — Hue, Saturation & Brightness used for color correction work.

Sharpening is to highlight transitions in intensity.

Thresholding is a useful technique for establishing boundaries in images that contain solid objects resting on a contrasting background.

There exist a large number of gray-level based segmentation methods using global or local image info.

The threshold technique requires that an object has homogeneous intensity and a background with a different intensity level.

Discrete Sequence of intensity values f: $x(n) = [4, 2, 1, 5]$.
Find the corresponding DFT.

DFT function,

$$F(m) = \sum_{0}^{M-1} f(n)\, e^{-J2\pi m n/M}$$

$$F(0) = \sum_{0}^{3} f(n)\, e^{-J2\pi \cdot 0 \cdot n/4} = \sum_{0}^{3} f(n)$$

$$= f(0) + f(1) + f(2) + f(3).$$

$$= 4 + 2 + 1 + 5.$$

$$= 12.$$

$$F(1) = \sum_{0}^{3} f(n)\, e^{-J2\pi n/4} = \sum_{0}^{3} f(n)\, e^{-J\pi n/2}$$

$$= f(0) + f(1)e^{-\pi J/2} + f(2)e^{-\pi J} + f(3)e^{-3\pi J/2}$$

$$= 4 + 2(\cos\tfrac{\pi}{2} - J\sin\tfrac{\pi}{2}) + (\cos\pi - J\sin\pi) + 5(\cos\tfrac{3\pi}{2} - J\sin\tfrac{3\pi}{2}).$$

$$= 4 + 0 - J2 + 0 - 1 + 0 + 5$$

$$= 8 - 2J$$

$$F(2) = \sum_{0}^{3} f(n)\, e^{-J\pi n}$$

$$= f(0) + f(1)e^{-J\pi} + f(2)e^{-2\pi J} + f(3)e^{-3\pi J}$$

$$= 4 + 2(\cos\pi - J\sin\pi) + (\cos 2\pi - J\sin 2\pi) + 5(\cos 3\pi - J\sin 3\pi)$$

$$= 4 - 2 + 1 - 5 = -2$$

$$F(3) = \sum_{0}^{3} f(n)\, e^{-J3\pi n/2}$$

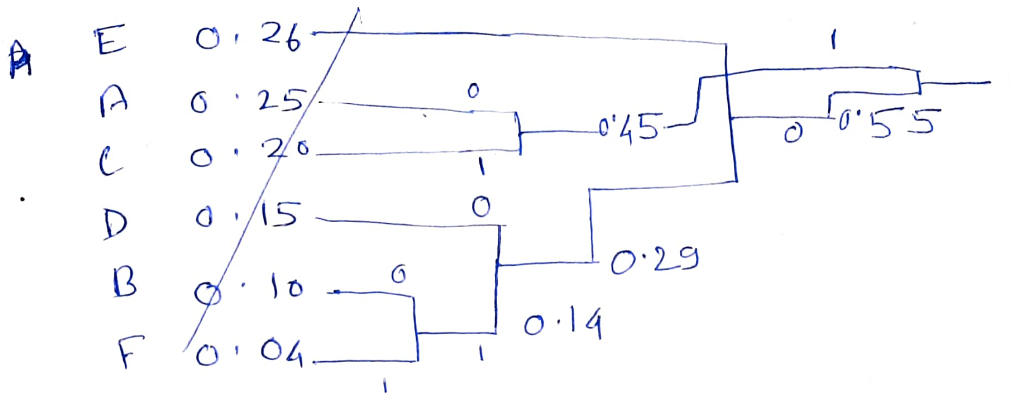
$$= f(0) + f(1)e^{-J3\pi/2} + f(2)e^{-J3\pi} + f(3)e^{-J9\pi/2}$$

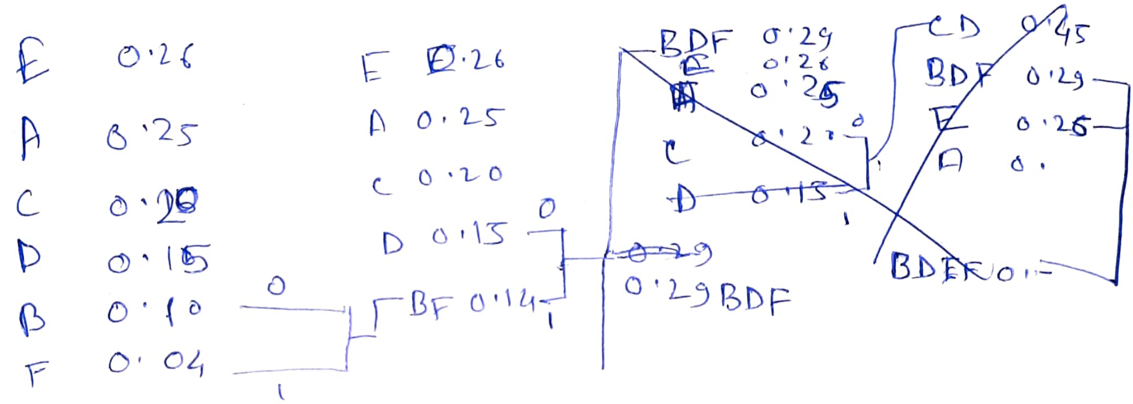
$$= 4 + 2(\cos\tfrac{3\pi}{2} - J\sin\tfrac{3\pi}{2}) + 1(\cos 3\pi - J\sin 3\pi) + 5(\cos\tfrac{9\pi}{2} - J\sin\tfrac{9\pi}{2})$$

$$= 4 + 2J - 1 + 5\cancel{J} = 8 + 2J$$

DFT of $x(n) = [12, 8-2J, -2, 8+2J]$.

g(b).





BDF 0·29

Group-A

1. A)
```
int i;
int ~~sum=0;~~ j=5;
~~for(i=0~~ int sum=0;
while (sum <10)
{
    sum=sum+i*j;
    i++;
}
p("    ",sum,i,j);
```
sum=0

1*5

```
int j=5;
int sum,i;
for (sum=0; sum<10000; sum=sum+i*j)
{ i++;
}
printf("%d %d \n",sum,i,j);
```

B)
a) break → valid
b) -1 → invalid
c) while → valid
d) int ^jet → invalid

2+2.5*2

c)

```
main()
{
printf("Expression values=%d %d \n", 5/2*2, 2+5/2*3-1);
}
```

output - Expression values = 5

D) 99+99 = 198 time

i) select name from Loan;
select name from Deposit;

ii) select name from Loan;

iii) select name from Deposit where branch = "xyz" and balance > 7500.

First Order Derivative filters. ~~Test~~

1) Roberts Operator (Cross Gradient) $\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y)$

$g_x = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$ $g_y = \begin{matrix} 0 & -1 \\ 1 & 0 \end{matrix}$ 2x2 Mask

$\begin{matrix} Z_1 & Z_2 & Z_3 \\ Z_4 & Z_5 & Z_6 \\ Z_7 & Z_8 & Z_9 \end{matrix}$

Image

Detect edges towards diagonal

2) Sobel

$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$ $\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$

More emphasis that are close to center of the mask and diagonal is only 1

3) Prewitt Op

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

used to detect vertical or horizontal Pixels in images.

Problems with Roberts

1) 2x2 masks are not easy to implement.
2) No. of Calculation are more.
3) No. of neighbouring Pixels considered in one go are less.

To improve / Solve

1) change in the size of mask
2) Change in the no. of neighbouring pixels.

1) Apply Robert, Sobel and Prewitt operator on the Pixel (1,1)

$$\begin{matrix} 50 & 50 & 100 & 100 \\ 50 & \boxed{50} & 100 & 100 \\ 50 & 50 & 100 & 100 \\ 50 & 50 & 100 & 100 \end{matrix}$$

Input

we need to traverse across the pixel (1,1)

$$\begin{matrix} 50 & 50 & 100 & 100 \\ 50 & \boxed{50} & 100 & 100 \\ 50 & 50 & 100 & 100 \\ 50 & 50 & 100 & 100 \end{matrix}$$

1) Robert operator

Mask

Origin ←
$$\begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$$

$50 \times (-1) + 100 \times 1$

$= -50 + 100 = 50$

2) Sobel operator

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

$50(-1) + 50(-2) + 100(-1)$
$+ 50(1) + 50(2) + 100(1)$

$= -50 + (-100) + (-100) + 50$
$+ 100 + 100$

$= 0$

3) Prewitt

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

$50(-1) + 50(-1) + 100(-1) +$
$1(50) + 1(50) + 1(100)$

$= 0$

First Order

Objective is to measure the intensity gradient.
Edge detectors can be vied as gradient calculators.

An edge can be extracted by computing the derivative of the image function.

(1) magnitude of the derivative which indicates the Strength or contrast of edge

(2) Direction of the derivative vector, indicates the edge orientation

Backward diff

2 ——— 1

$x-\Delta x$ ——— x

$$= [f(x) - f(x-\Delta x)] / \Delta x$$

Forward

1 ——— 2

x ——— $x+\Delta x$

$$= [f(x+\Delta x) - f(x)] / \Delta x.$$

Central Difference

2 ——— 1 ——— 0

$x-\Delta x$ ——— x ——— $x+\Delta x$

$$= [f(x+\Delta x) - f(x-\Delta x)] / 2\Delta x$$

These differences can be obtained by applying the following masks, assuming $\Delta x = 1$

BD $\rightarrow$ $f(x) - f(x-1)$

$= [1 \quad -1]$

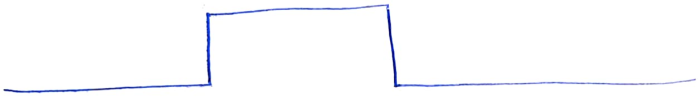
FD — $f(x+1) - f(x)$

$= [-1 \quad 1]$

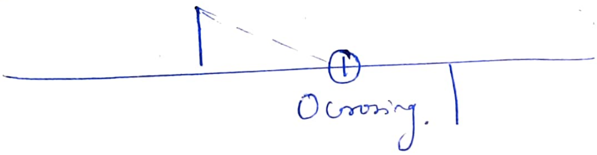
2nd order derivative

1) First derivative edges are considered to be present when edge magnitude is larges compared to threshold value.
2) In case of 2nd, edge is present at that location when the 2nd derivative is zero.
3) It is like 0 crossing which can be observed as a sign change.

1st



2nd



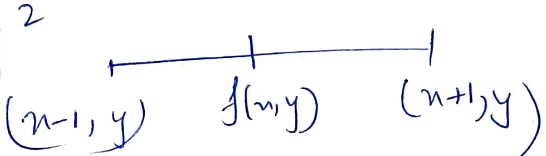
Laplacian algo.

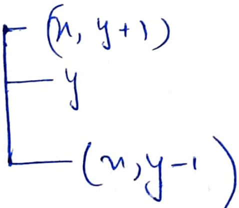
1) 0 crossing algo, No magnitude check.
2) This mask is very sensitive to noise even a small ripple looks like edge
3) So image must be filtered first and then edge detection is applied.

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

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(x+1,y) - 2f(x,y) + f(x-1,y)$$

$$\frac{\partial^2 f(x,y)}{\partial y^2} = f(x,y+1) - 2f(x,z) + f(x,y-1)$$





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

Algo

1) Generate the mask
2) Apply the mask
3) Detect the 0 crossing.

0 crossing.

p & q $\rightarrow$ $|\nabla^2 f(p)| \leq |\nabla^2 f(q)|$

0	-1	0		-1	-1	-1
-1	4	-1		-1	8	-1
0	-1	0		-1	-1	-1

only center coefficient has +ve value.

3x3 mask.

1) $\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$

2) $\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$.

0	1	0		1	1	1
1	-4	1		1	-8	1
0	1	0		1	1	1

D -> 1, 5, 7, 12, 17, 18, [illegible], 32, 34, 35, 36, 39, 43, 48, 46, 50, 53, 58, 55, 59, 60, 81, 62, 67, 66, 75 79, 80, 72, 61, 78,

AIML – 27, 12, 23, 21, 24, 71, 38, 1, 66, 67, 54, 69, 29, 52, 50, 59, 63, 72, 7 25, 60, 68