INDEX

NAME: SWAGAT PANDA STD.: SEC: ROLL NO.: SUB: DIGITAL IMAGE PROCESSING

Text book: Gonzalez & Woods . Digital Image Processing

Reference books: i) Sonka, Vaclav & Boyle , Image Processing , Analysis & Machine vision.

ii) Anil K. Jain, Fundamentals of Digital Image Processing

iii) Gonzalez, Woods & Eddins, Digital Image processing using MATLAB.

Image → A subset of signal (1D, 2D, 3D)

An image is a function of spatial parameters that conveys information about an object or it's attributes.

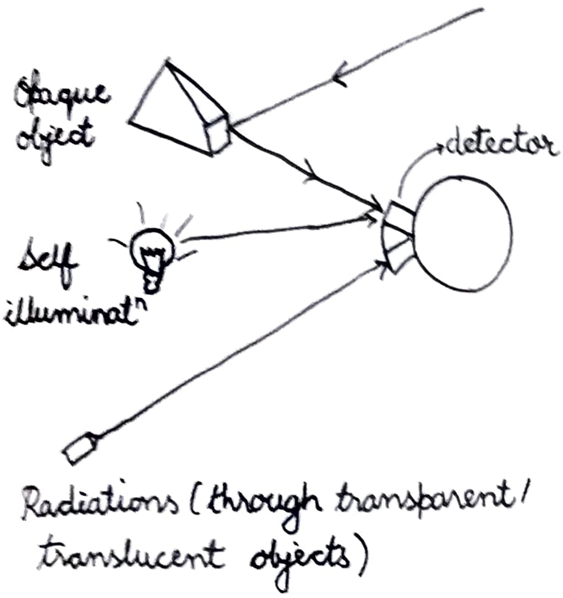
Defects to be corrected in a bad image →

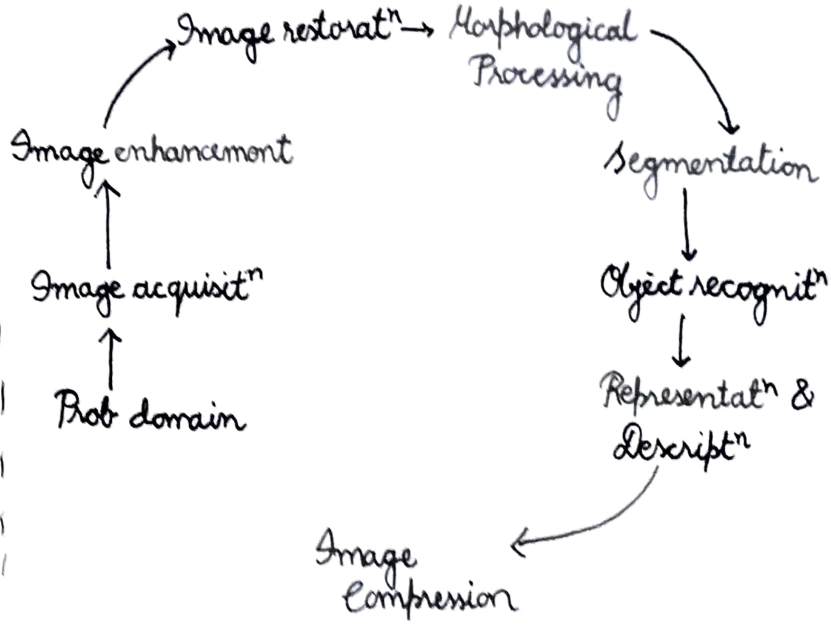
Colour Fringing, Flash defect, JPEG artefacts

Noise, Blur, Colour Imbalance

Exposure problems.

A video: A set of images played in time





Evaluation Scheme

Component	Duration	Weightage	Date & Time	Nature of component
Test 1	50 min	15%		
Test 2	"	15%		
Test 3	"	15%		
Comprehensive exam	2h	35%		
Quiz / Assignment / MATLAB Coding	–	20%	Announced in class	Closed/Open

λ (in μm)	Visible	Near-IR	SWIR	MWIR	LWIR
	0.4-0.75	0.75-1.1	1-3	3-5	7-14

Scanning an image in different wavelengths → can help us get more info about a situatⁿ: such as forensic applicatⁿs, MRI & X-rays
→ Understanding & Interpreting them requires image processing

6/6 vision → Snellen fraction → means one can normally /clearly see at 6m for what should be seen at 6m.

Human eye & it's structure (Rods & Cones – cells)
b/w dim — color & bright

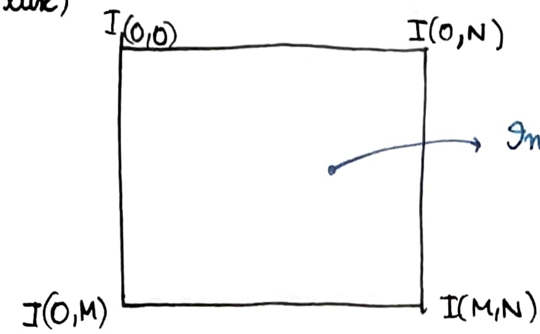
Visible range of light → 400–750nm

3 types of cones → R, B & G sensitive

Blind spot → base of the optic nerve.

30fps → video ; >30fps

Source of Light → incident light → reflected light → projection → sensor → electrical signal
(Et erit lux)



→ Intensity at (m,n) → Quantizatⁿ for a digital image

If image is b&w → I(m,n) is a single value in grayscale (2^8) {0 – 255}

If image is colored → I(m,n) → triplet vector
{H,S,V} → {Hue, Saturation & Value of Intensity}

Image Resolutⁿ

→ Spatial resolutⁿ
> 256×256 (pixel × pixel)
1024×1024
> 1250 DPI (dots per inch)

→ Bit resolutⁿ

→ Temporal resolutⁿ

The pixel density should be twice the highest spatial freq. in the image. In practice, 2.5 times is used to compensate for the dist. across the diagonal of the pixel

The Image Digitizatⁿ (mapping cont. signal to discrete pixels)
→ Spatial quantizⁿ
→ Gray level quantizⁿ

Nyquist sampling th. or Shannon's sampling th. is applicable in image (spatial frequency & not temporal frequency as in videos) Acquired info ≥ 2(freq)

Quantizatⁿ introduces a distortion into images called "aliasing"

Forms of aliasing → spatial, intensity or temporal

> Spatial Quant
→ Chess board effect

> Intensity Quant
→ False contouring effect

> Temporal Quant
No. of images captured in a given time period (fps)
E.g. Apparent backward motion of wagon wheels in western movies, because it's a sequence of still frames.

Image → gray image as a N×M matrix , $0 \leq N,M \leq 255$

matrix $A(i,j)$ $i = 1, \ldots, M$
$j = 1 \ldots N$

MATLAB IP functions → imread, imwrite, iminfo, imshow, imagesc, colormap

imfinfo ('   ')

IM = imread ('   ')

imwrite (IM, '   ')

imshow (IM);

imagesc(IM);

axis image;

axis off;

colormap.

Fan speed >30fps → reversed direction
(human brain can process only 30fps)

Aliasing
(not enough sampling)

Image of ramp

```
for i = 1:256
   for j = 1:256
      A(i,j) = j-1
   ~~end~~
end
 image(IM)
colormap( gray(256));
axis('image')
```

Image of circle

```
for
   for
      dist =((i-128)^2+ (j-128)^2)^(1/2)
      if(dist<80)
         B(i,j) =255;
      else
         B(i,j)=0;
      end
   end
end
image(B);
colormap( gray (256))
axis('image')
```

Weber Ratio
Sampling & Quantization, Brightness Adaptatⁿ & Discrimination Isopreference curve
Basic relationship b/w pixels & Interpolation

DIY → Take your picture → Change resolutions of the image
Display image, obtain printout in pdf
Convert your image to 8 bit gray shade

Brightness Adaptatⁿ & Discriminatⁿ

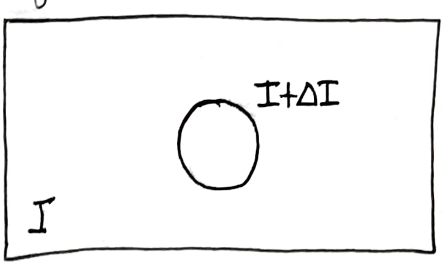
→ Range of light intensity levels to which HVS can adapt - order of 10^{10}
→ Intensity as percieved by HVS → a log fⁿ.
→ HVS cannot operate over such a range simultaneously
→ Current sensitivity level of human eye → brightness adaptatⁿ level.

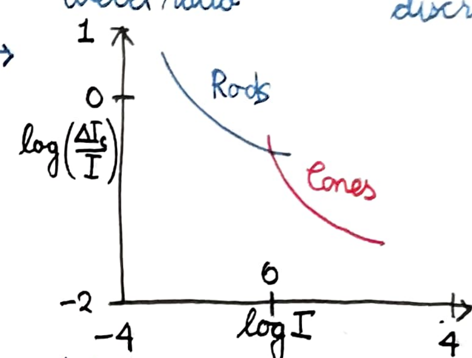
Weber's law → 'The ratio of the increment threshold to the background intensity is a const'.

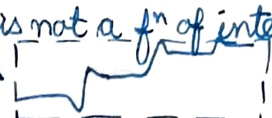
The eye discriminates b/w level changes in brightness at any specific adaptation level by Weber ratio:

$$\boxed{\text{Weber ratio} = \frac{\Delta I_c}{I}}$$

Basic exptᵃˡ setup used to characterize brightness discriminatⁿ → Small values of Weber ratio → Good brightness discrimination



→


→ Percieved brightness is not a fⁿ of intensity
↓ → Mach bands →


↓ → Simultaneous Contrast
Brightness is subjective matter

→ A typical observer → 12-24 different intensity changes; i.e the no. of different intensities a person can see @ any one pt. in a monochrome image

→ Overall intensity discriminatⁿ is broad due to different set of incremental changes to be detected at each new adaptatⁿ level

Vary both sampling & Quantization

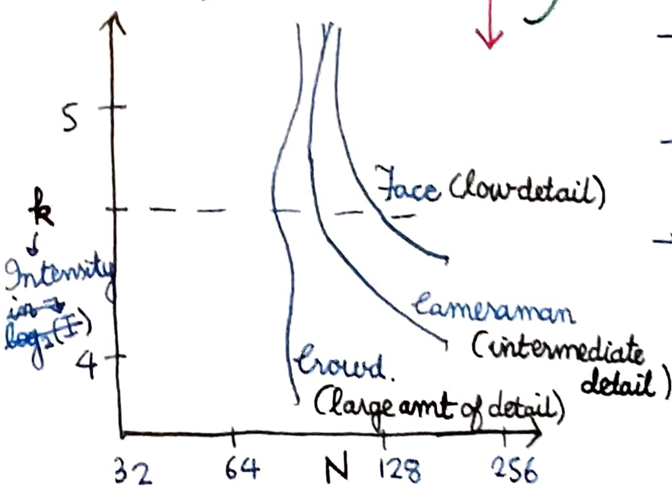
Mapping of continuous signals to a discrete no. of spatially organized pts. (pixels)

→ Spatial Quant & Chessboard effect → Gray level quantizatⁿ & contour effect

Intensity of k bits for N pixels

Iso-preference curves

Points on this correspond to images of equal subjective quality



Remarks

→ Isopreference curves tend to shift right & upward. (i.e. better image quality)

→ In images with a large amt. of detail only a few gray levels are need.

→ In the other two image categories, the perceived quality remained the same in some intervals in which N was inc but k actually dec.

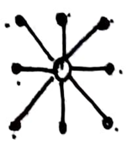
Zooming, Shrinking

$\underbrace{\text{Zooming , Shrinking}}_{\text{Using digital interpolations}}$

Interpolⁿ → Process of using known data to estimate values at unknown locations data.

Neighbors of pixels & Adjacency

$N_4(p) = \{(i-1,j), (i+1,j), (i,j-1), (i,j+1)\}$



$N_8(p) = \{(i+1,j+1), (i,j+1), (i+1,j), (i+1,j-1), (i-1,j+1), \cancel{(i,j)}, (i-1,j-1), (i,j-1), (i-1,j)\}$

Distance measures D

a) $D(p,q) \geq 0$ ($D(p,q) = 0$ when $p = q$)

b) $D(p,q) = D(q,p)$

c) $D(p,z) \leq D(p,q) + D(q,z)$

Examples

- Euclidean: $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$
- D_4 (city block or Manhattan) ⇒ $D_4(p,q) = |x-s| + |y-t|$
- D_8 (chessboard) ⇒ $D_8(p,q) = \max\{|x-s|, |y-t|\}$

Interpolatⁿ → Shrinking, zooming, rotating & geometric correction

How to interpolate? →

↳ Linear Interpolation → Val of I $f(x,y)$ at $P(x,y)$

↓

Assume we know

$f(x_1,y_1) = Q_{11}$ &

$F(x_1,y_1) = aQ_{11}$ (Intensity)

$f(x_2,y_1) = Q_{21}$ &

$F(x_2,y_1) = bQ_{21}$ (Intensity)

↳ Nearest Neighbour

↳ Bilinear

Decrease in 'k' tends to Increase the apparent contrast

↳ which is why Lena & photographer's images have dec in gray levels as quality inc

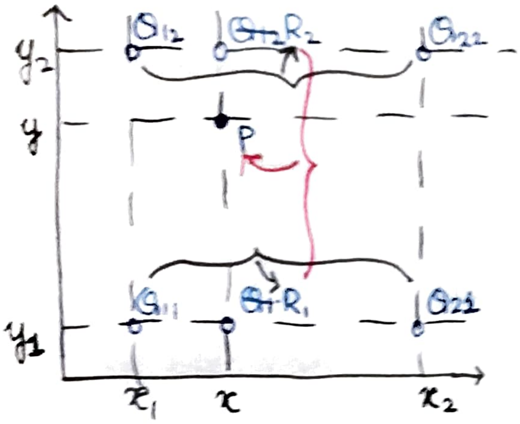
Interpolⁿ

Pixels with no values when an image is enlarged → their grayscale values have to be interpolated such that the image is enlarged properly.
↳ Reqd to reconstruct a continuous signal

$4\times4 \xrightarrow[400\%]{\text{enlarge by}} 8\times8$ (contains only 25% info when its not interpolated)

Bilinear interpolation (Q → intensity)

$$P = (1-c)(1-r)\,Q_{12} + (1-c)r\,Q_{22} + c(1-r)\,Q_{11} + cr\,Q_{21}$$



$R_1 = (1-r)\,Q_{11} + r\,Q_{21}$

$R_2 = (1-r)\,Q_{12} + r\,Q_{22}$

using R_1 & R_2

$P = (1-c)R_2 + cR_1$

Ratios of dist b/w the pts
r = row
c = columⁿ

Example

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

Assume $c = r = 0.5$

$\xrightarrow{8\times8}$

2	1.5	1	2.5	4	3	2	–
2.5	1.75	1	2	3	2.25	1.5	–
3	2	1	1.5	2	1.5	1	–
2	1.75	1.5	1.5	1.5	2	2.5	–
1	1.5	2	1.5	1	2.5	4	–
2.5	2	1.5	1.5	1.5	2.5	3.5	–
4	2.5	1	1.5	2	2.5	3	–
–	–	–	–	–	–	–	

→ This is a simple & computationally efficient.

Advanced, e.g. → bicubic & B-spline, edge orientatⁿ based interpolatⁿ, transform domain etc.

Zoom in, Zoom out

Scaling factor > 1 → Zoom out Scaling factor < 1 → Zoom in.

→ For binary images → just replicate pixels while scaling up.
→ downscale by simple sub sampling.

Interpolatⁿ artifacts

Scale factor > 1 → Upscaling Scale factor < 1 → Downscaling

→ Downscale: simple subsampling & take chances with aliasing.

→ Apply a lowpass filter before subsampling

→ Digital zoom & Optical zoom

Image $\xrightarrow{\text{Upscaling}}$ Image

↳ Problems → Aliasing, Blurring, Edge halo

MATLAB functions

imresize() → Image resize (interpoln is used)

tic() / toc() functⁿ → calculates computation time

Interpolation methods

Methods:
- 'nearest'
- 'bilinear'
- 'bicubic'

Kernels:
- 'box'
- 'triangle'
- 'cubic'
- 'lanczos2'
- 'lanczos3'

Application of Interpolatⁿ

Super resolution img reconstructⁿ → Produces high res. img, using the existing low-cost img devices from a single image or few snapshots of low res imgs.

Image Enhancement

Process of making images more suitable for a particular applicatⁿ

- → Highlight details → highlighting edges
- → Visual appeal inc. → improve contrast
- → Sharpening / de blurring → removing noise

Enhancement of an image is very subjective & domain dependent, & not generic at all. (different for each image)

→ Digital data is rigid & once acquired, ~~and once acquired~~ it cannot be perfected with more info than it had originally.

Unnoticed details ~~Enhancements~~ are present in ~~the~~ image & they are enhanced by removal of noise. The nature of noise is usually additive.

NO extra visual info is added during enhancement

Digital image enhancements techniques suppress noise, blur, uneven contrast or motion & other artifacts & focus on details.

Methods of enhancement

→ Spatial Domain (direct manipulatⁿ) → Transform domain (FT)

→ Hybrid (FT + direct)

Spatial domain techniques

a) Single - pixel operation. (Pt transform)

b) Neighbourhood operations (Local t/f)

c) Global operation (Global transform)

PIXEL LEVEL PROCESSING

Point Transformation : The o/p value at a specific coordinate is dependent only on the i/p value at the same coordinate

Local Transformation : The o/p value at a specific coordinate is dependent on the neighbourhood of the same coordinate.

Global Transformation: The o/p value at a specific coordinate is dependent on all values in the i/p image.

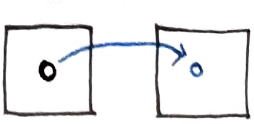
a) Single Pixel ~~Processing~~ Operatⁿ → Scaling, histogram manipulatⁿ, thresholding

b) Neighbourhood operatⁿ → ~~Neighbourhood op~~: edge detection, smoothing

c) Geometric spatial t/f → ...

d) Global operations. → Fourier, Hough, Principal component wavelet

PIXEL LEVEL PROCESSING



Pt t/f

Local t/f

Global t/f

Interpolatⁿ artifacts

Scale factor > 1 → Upscaling Scale factor < 1 → Downscaling

→ Downscale: simple subsampling & take chances with aliasing

→ Apply a lowpass filter before subsampling

→ Digital zoom & Optical zoom

Image $\xrightarrow{\text{Upscaling}}$ Image

↳ Problems → Aliasing, Blurring, Edge halo

MATLAB functions

imresize() → Image resize (interpolⁿ is used)

tic() / toc() functⁿ → calculates computation time

Interpolation methods

'nearest', 'bilinear', 'bicubic' } Methods

Kernels { 'box', 'triangle', 'cubic', 'lanczos2', 'lanczos3'

Application of Interpolatⁿ

Super resolution img reconstructⁿ → Produces high res. img, using the existing low-cost img devices from a single image or few snapshots of low-res imgs.

Image Enhancement

"Process of making images more suitable for a particular applicatⁿ"

→ Highlight details → highlighting edges
→ Visual appeal inc. → improve contrast
→ Sharpening / deblurring → removing noise

Enhancement of an image is very subjective & domain dependent, & not generic at all. (different for each image)

→ Digital data is rigid & once acquired, ~~and once acquired~~ it cannot be perfected with more info than it had originally.

Unnoticed details ~~Enhancements~~ are present in ~~the~~ image & they are enhanced by removal of noise. The nature of noise is usually additive.

NO extra visual info is added during enhancement

Digital image enhance-ments techniques supp-ress noise, blur, uneven contrast or motion & other artifacts & focus on details.

Methods of enhancement

→ Spatial Domain (direct manipulatⁿ) → Transform domain (FT) → Hybrid (FT + direct)

Spatial domain techniques

a) Single - pixel operation. (Pt transform) b) Neighbourhood operations (Local t/f)

c) Global operation (Global transform)

PIXEL LEVEL PROCESSING

Point Transformation : The o/p value at a specific coordinate is dependent only on the i/p value at the same coordinate

Local Transformation : The o/p value at a specific coordinate is dependent on the neighbourhood of the same coordinate.

Global Transformation: The o/p value at a specific coordinate is dependent on all values in the i/p image.

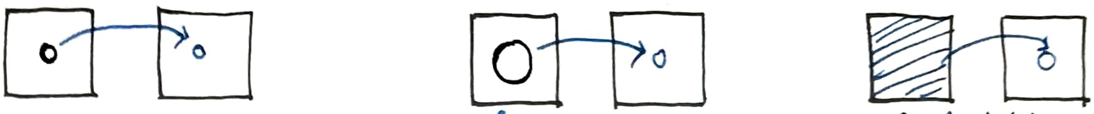
a) Single Pixel ~~Processing~~ Operatⁿ → Scaling, histogram manipulatⁿ, thresholding

b) Neighbourhood operatⁿ → ~~Neighbourhood op~~: edge detection, smoothing

c) Geometric spatial t/f → ...

d) Global operations. → Fourier, Hough, Principal component wavelet

PIXEL LEVEL PROCESSING



Pt t/f Local t/f Global t/f

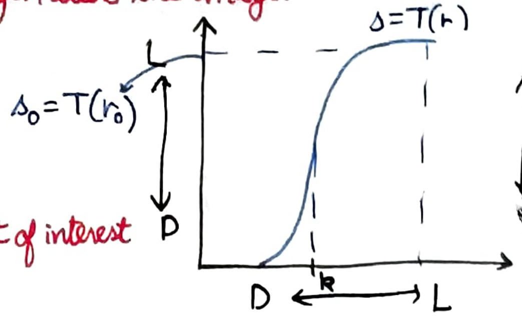
Negative Image: $\mathcal{F}(x,y) = 255 - f(x,y)$

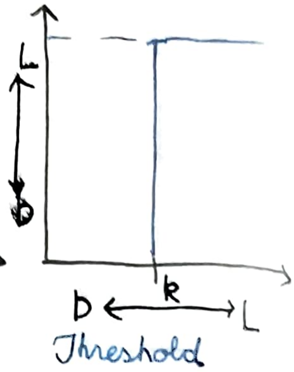
Brightness Adjustment: $\mathcal{F}(x,y) = C + f(x,y)$; $C \rightarrow$ Intensity (DC level analogy)

↳ For a 8 bit image if we don't take care of the limit of 255 max size, then there's an overflow & it damages/alters the image.

Intensity transformation:

$s = T(r)$

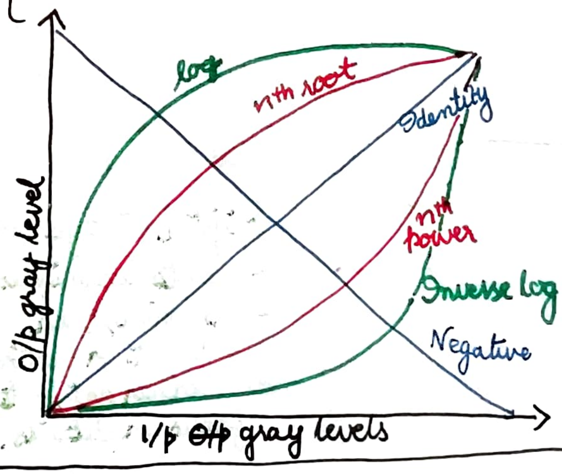




$$\boxed{\text{Threshold} \rightarrow \begin{array}{ll} s = 1 & ; r > k \\ s = 0 & ; r < k \end{array}}$$

Useful for segmentatⁿ in which an object of interest is isolated from the background.

Gray level transformation



Log transformation

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

Power law, γ transform

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

$\gamma < 1 \rightarrow$ darker details are revealed (image gets lighter)

$\gamma > 1 \rightarrow$ lighter details are revealed.

Gamma correction is required so that the image appears as seen by the eye & not a digital camera.

Point processing techniques

- → Negative Images
- → Brightness Adjustment
- → Thresholding Images
- → Logarithmic t/f
- → Power law t/f
- → Grey level slicing
- → Bit plane slicing

→ Log transformations are particularly useful when the i/p grey level values have an extremely large range of values.

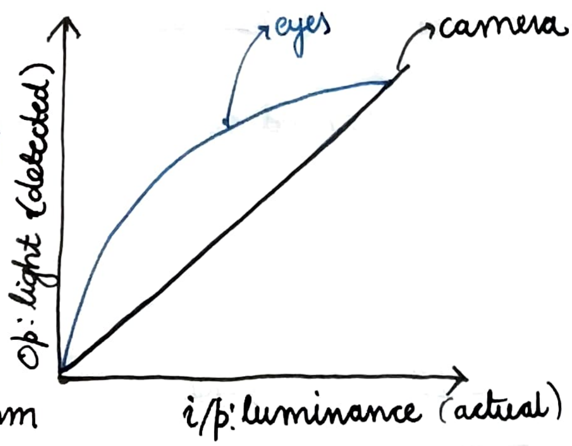
→ Inverse log transformatⁿ does the opposite → enhances the lighter parts varyingly keeping darker parts const.

Gamma correction

→ Our eyes don't percieve light the way cameras acquire it.

→ Therefore monitors & displays use γ correction s.t. it appears natural to the human eye

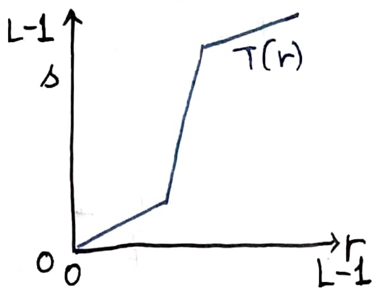
→ It can be corrected to a log transform



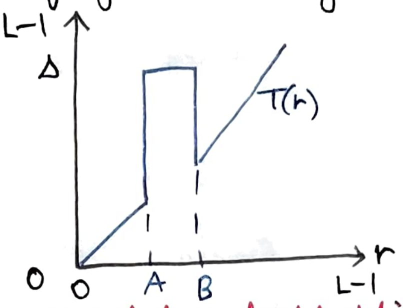
$\boxed{31/08/20}$ - L6

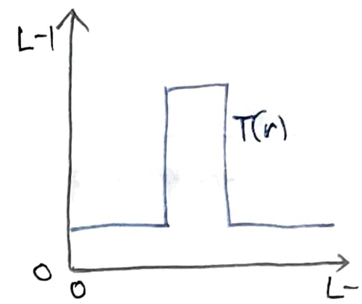
Piecewise Linear Transforms

Using piecewise f^ns to enhance certain intensities & not using absolute thresholds.



→ Gray level slicing

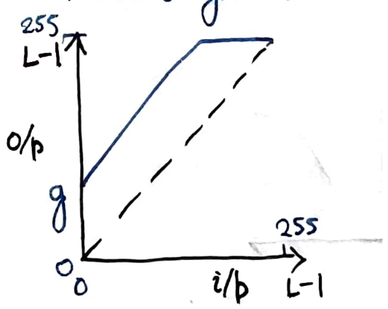




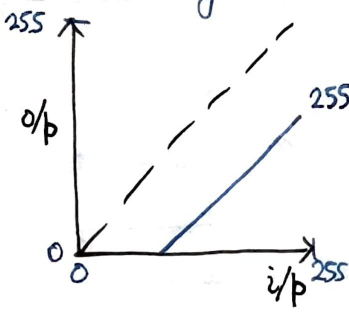
→ useful in highlighting features of an image.

Point processing : Image editing

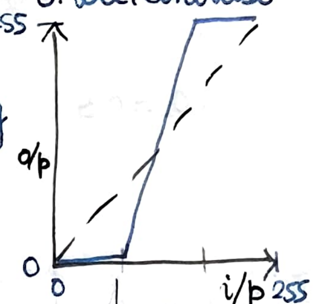
1. Inc brightness



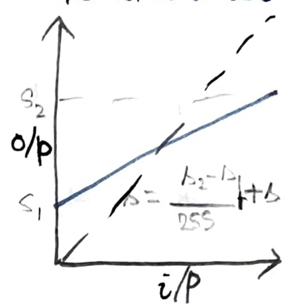
2. Dec. brightness



3. Dec. Contrast



4. Inc. Contrast



$s = \frac{s_2 - s_1}{255} r + s_1$

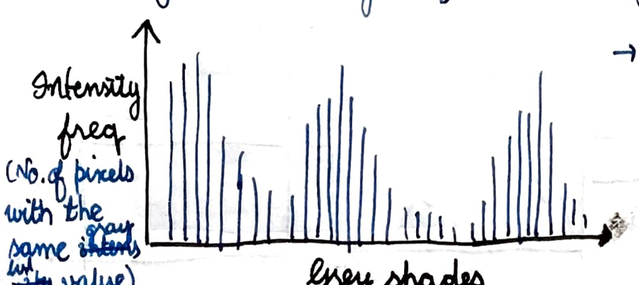
$s = \left(\frac{255 - 0}{r_2 - r_1}\right) r + C$

$0 = \left(\frac{255}{-r_1 + r_2}\right) r_1 + C \qquad C = \frac{-255 r_1}{-r_1 + r_2}$

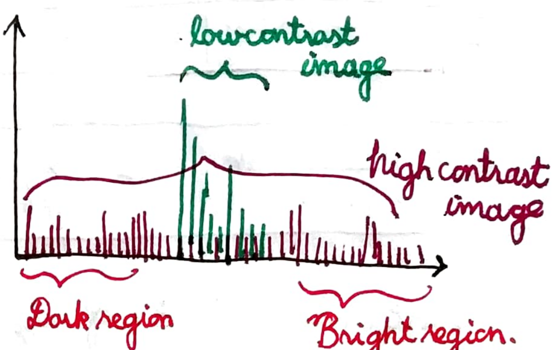
Info is lost by Image editing!

Image histograms :

The histogram of an image shows us the distribut^n of gray levels in the image. Massively useful in image processing.



→ A good image



High contrast image : best quality, evenly distributed histogram

Histogram equalization

→ Based on mathematical f^n & simple to implement & acceptable approach for enhancement

→ Automatically determines a tranfer function T.

If we reshuffle all pixels within the image, ~~but~~ it doesn't change the histogram but destroys the image.

→ Images contain a lot of spatial information & it is important.

→ Histogram is a global descriptor & reflect no structural or spatial info of pixel

→ If we apply histogram equalization then pixels' spatial location should be preserved

An image whose pixels tend to occupy the entire range of possible gray level values & tends to distributed uniformly, will have a high contrast and a range (wide) of gray tones.

→ We can enhance the images that have a poor contrast by applying a pretty simple contrast specificatⁿ (Histogram equalizer)

BASIC RULES

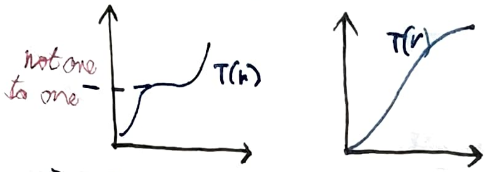
Transformations (intensity mappings) of the form: $s = T(r) \quad 0 \le r \le 1$ that produce an output intensity level 's' for every pixel in the i/p image having intensity 'r'.

Assume that:

(a) $T(r)$ is single valued & monotonically increasing f^n in the interval $0 \le r \le 1$ &;

↳ important for one to one relation, mapping

(b) $0 \le T(r) \le 1$ for $0 \le r \le 1$



Algorithm

The gray levels in an img maybe viewed as a random variable. The fundamental descriptor of random variable is the probability density function.

The PDF of ~~a~~ the transformed variable 's' can be obtained by.

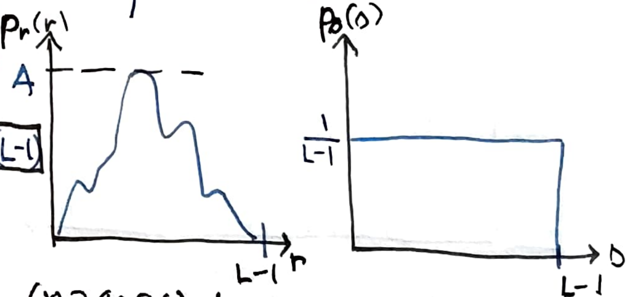
$$\boxed{\begin{gathered} p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \\ s = T(r) = \int_0^r p_r(w).dw \end{gathered}}$$

Now by Lebinitz rule: $\frac{ds}{dr} = \frac{dT}{dr} = \frac{d}{dr}\left(\int_0^r p_r(w).dw\right)$ · then I →

I → $\boxed{p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1 \text{ for } 0 \le s \le 1}$

"Histogram equalizatⁿ transformatⁿ":

$$\boxed{s_k = \int \frac{p_r(r)}{p_s(s)}.dr = (L-1)\sum_{j=0}^{k}\frac{n_j}{n}} \quad \boxed{\text{for } k = 0, 1, \ldots (L-1)}$$



3 bit image: (L=8) of size 64×64 pixels (n=4096) has the intensity dist.s

→ Applying histogram equalizⁿ transform

r_k	n_k	$p_r(r_k) = n_k/MN$	r_k
$r_0 = 0$	790	0.19	r_0
$r_1 = 1$	1023	0.25	r_1
⋮	850	0.21	r_2
⋮	656	0.16	⋮
⋮	329	0.08	⋮
⋮	245	0.06	⋮
⋮	122	0.03	⋮
$r_7 = 7$	81	0.02	r_7

Here $p_r(r) = \frac{n_r}{n}$; $p_s(s) = \frac{1}{L-1}$

$$\Rightarrow s_k = (L-1)\sum_{j=0}^{k}\frac{n_j}{n}, \quad k = 0, 1, \ldots L-1$$

$\boxed{01/09/20}$ — L7

→ Recap of histogram equalizatⁿ algorithm

Intensity values

r_k	n_k	$P_r(r_k) = n_k/MN$	CDF
$r_0 = 0$	790	0.19	→ 0.19 × 7 = 1.4
$r_1 = 1$	1023	0.25	→ 0.44 × 7 = 3.08
$r_2 = 2$	850	0.21	→ 0.65 × 7 =
$r_3 = 3$	656	0.16	0.81 × 7 =
$r_4 = 4$	329	0.08	0.89 × 7 =
$r_5 = 5$	245	0.06	0.95 × 7 =
$r_6 = 6$	122	0.03	0.98 × 7 =
$r_7 = 7$	81	0.02	1 × 7 =

s values ↓

→ $\frac{1}{7} = 0.142$

$\frac{2}{7} = 0.285$

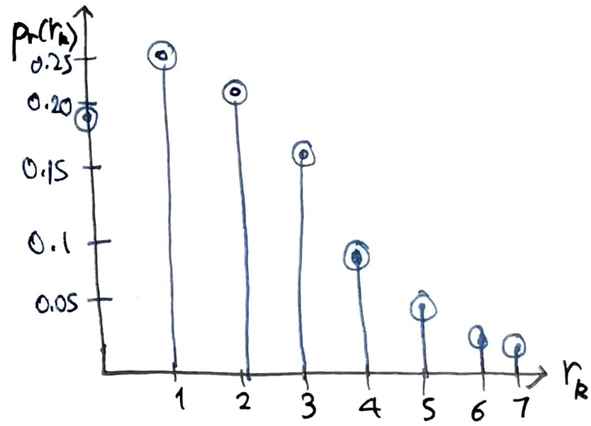
$\frac{3}{7} = 0.426$

$\frac{4}{7} = 0.568$

$\frac{5}{7} = 0.71$

$\frac{6}{7} = 0.856$

$1 = 1$

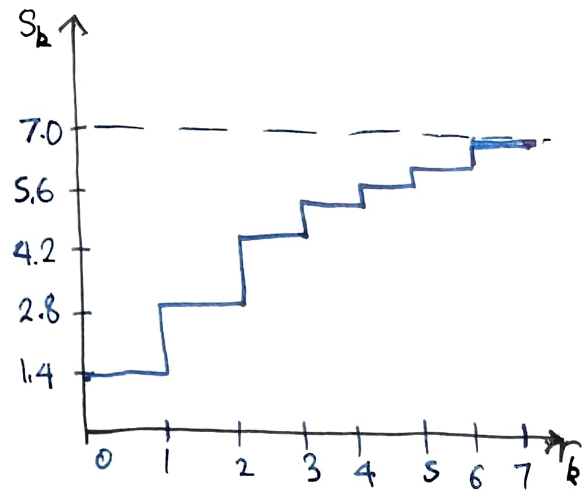


Processed pixel intensity value:

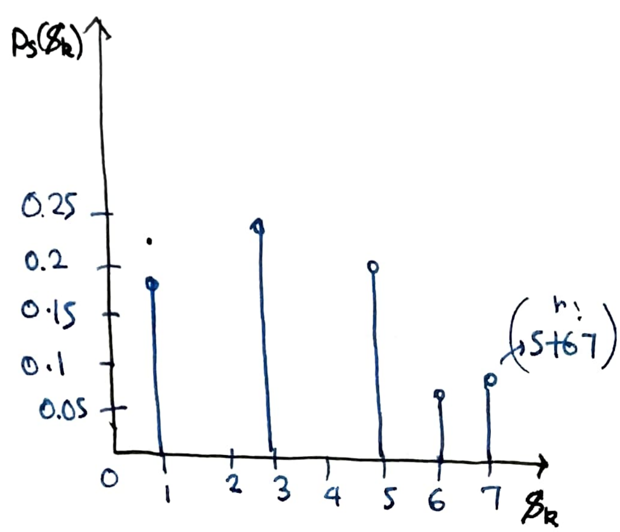
$$s_k = (L-1)\sum_{j=0}^{k} \frac{n_j}{n} \; ; \; \text{for } k = 0, 1, 2 \ldots (L-1)$$

$S_0 = 1.33 = 1$; $S_2 = 4.55 = 5$; $S_4 = 6.23 = 6$; $S_6 = 6.86 = 7$;
$S_1 = 3.08 = 3$; $S_3 = 5.67 = 6$; $S_5 = 6.65 = 7$; $S_7 = 6.86 = 7$;

$$\left.\begin{matrix} S_0 = 1 & S_2 = 5 & S_4 = 6 & S_6 = 7 \\ S_1 = 3 & S_3 = 6 & S_5 = 7 & S_7 = 7 \end{matrix}\right\} \text{Mapping}$$



r		s
0	⟶	1
1	⟶	3
2	⟶	5
3	⟶	6
4	⟶	6
5	⟶	7
6	⟶	7
7	⟶	7



Histogram matching / Specification

(For local enhancement, we have to modify histogram on the basis of local requirements or specifications.)

→ Histogram matching / specificatⁿ enables us to "match the grayscale distribution in one image to the grayscale distribution in another image.

> Given images – i/p A & desired B, using pt processing we would like to generate an image C from a such that $p_C(i) \sim p_B(i)$; $i = 0, \ldots 255$.

> More generally, given an image A & a histogram $h_B(i)$, we would like to generate an image C s.t. $p_C(i) \sim p_B(i)$; ($i = 0, \ldots 255$)

i/p transform : $s = T(r) = \int_0^r p_r(w)\,dw$; $p_r(r)$ for a given image

o/p desired : $G(z) = \int_0^z p_z(t).dt$, $p_z(z)$ for desired image

$s = G(z)$

Inverse transform to get output for specified histogram :

$$G(z) = T(r) = s \Rightarrow z = G^{-1}(s) = G^{-1}(T(r))$$

Example 3.7 TB> Assume continuous intensity values, suppose that an image has intensity PDF $p(r) = 2r/(L-1)^2$ & $p(r) = 0$ for other values. Find T(r) which will produce an image whose intensity PDF is $p(z) = 3z^2/(L-1)^3$. & $p(z) = 0$ for other values of z.

$$s = T(r) = (L-1)\int_0^r p(w).dw = \frac{r^2}{(L-1)} \cancel{= G(z)}$$

$$G(z) = (L-1)\int_0^z p(w).dw = \frac{z^3}{(L-1)^2} = G(z) \Rightarrow z = G^{-1}(s)$$

$$G^{-1}(s) = z = ((L-1)^2 \cancel{z} s)^{1/3} \Rightarrow z = \left[(L-1)^2 \frac{r^2}{(L-1)}\right]^{1/3} = \left[r^2(L-1)\right]^{1/3}$$

Procedure

1 → Obtain $s = T(r)$ using histogram equalization ~~to be specified histogram~~

2 → ~~For eve~~ Obtain $G(z)$ by histogram equalizing to specified histogram.

3 → For every value of s_k, $k = 0, 1, \ldots L-1$, use stored values of G from 2 such that corresponding value of z ~~is~~ gives $G(z)$ closest to s_k & store these mapping from s to z.

4 → Obtain the output image by applying $G(z)$ to all pixels in i/p image

Obtaining analytical expressions for T(r) & G(z) is hard

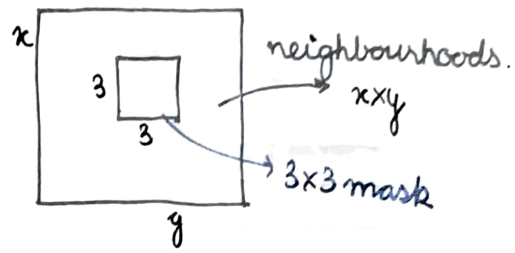
Check example 3.8 from textbook

Complete > Page 133

> ## Fundamentals of spatial filtering –

> One of the principal tools used in image processing for image enhancement
> It's mechanics consist of neighbourhood and a predefined operatⁿ are performed on neighbourhood pixels.
> A filtered image is generated as the center of the spatial filter moves across each pixel in the i/p image.
> Spatial filters can be used for linear & non linear filtering.

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



→ The neighbourhood of a pt f(x,y) can be defined using a square / rectangular (commonly used) or circular sub image area centered at (x,y).

→ The center of the sub image is moved from pixel to pixel starting at the top corner $g(x,y) = T(f(x,y))$

→ ~~where~~ Here f(x,y) is i/p image, g(x,y) is the processed image & T is an operator on f, defined over some neighbourhood of (x,y).

Spatial Correlation & Convolution

> Refer figure 3.29 ; pg 147 for conv. & correlations illustrated in a figure.
> For a 2D image → refer fig 3.30, pg 149

→ Spatial correlation is the process of moving a filter mask over the image & computing the sum of products at each locatⁿ

→ In spatial convolution, the filter is first rotated by 180° then moving over the image similar to spatial correlation & computing the sum of products at each locatⁿ.

Convolution is associated with filtering

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

The masks look like impulse responses.

Correlation is associated with pattern recognition

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

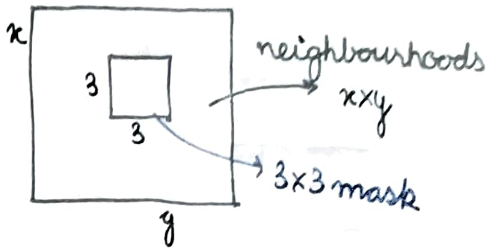
The masks look like objects to be searched for.

Check figures 3.28, 3.29 in the text: pg 146 – 147
" " 3.30 " " " : " 149

Fundamentals of spatial filtering

- One of the principal tools used in image processing for image enhancement
- It's mechanics consist of neighbourhood and a predefined operatⁿ are performed on neighbourhood pixels.
- A filtered image is generated as the center of the spatial filter moves across each pixel in the i/p image.
- Spatial filters can be used for linear & non linear filtering.

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



→ The neighbourhood of a pt f(x,y) can be defined using a square / rectangular (commonly used) or circular sub image area centered at (x,y).

→ The center of the sub image is moved from pixel to pixel starting at the top corner $g(x,y) = T(f(x,y))$

→ ~~w~~here f(x,y) is i/p image, g(x,y) is the processed image & T is an operator on f, defined over some neighbourhood of (x,y).

Spatial Correlation & Convolution

Refer figure 3.29 ; pg 147 for conv. & correlations illustrated in a figure.
For a 2D image → refer fig 3.30, pg 149

→ Spatial correlation is the process of moving a filter mask over the image & computing the sum of products at each locatⁿ

→ In spatial convolution, the filter is first rotated by 180° then moving over the image similar to spatial correlation & computing the sum of products at each locatⁿ.

Convolution is associated with filtering

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

The masks look like impulse responses.

Correlation is associated with pattern recognition

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

The masks look like objects to be searched for.

Check figures 3.28, 3.29 in the text: pg 146 - 147
" " 3.30 " " " : " 149

Smoothing Spatial filters

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(s+x, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)} = \frac{1}{9}\sum_{i=1}^{9} f_i$$

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

- → Noise reductⁿ
- → Reductⁿ of false contours
- → " " irrelevant details

Averaging filters (Low pass filter)

> Box filter

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

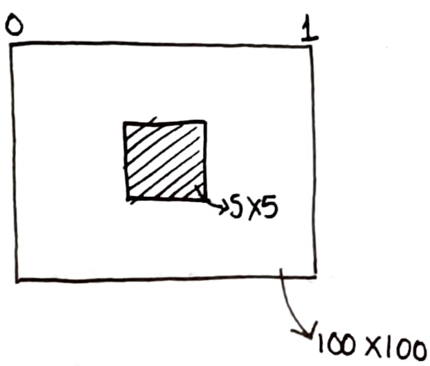
> Weighted avg filter

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

Smoothing $\propto$ Mask size.
More effect of smoothing with inc. in mask size.
Drawback of averaging is blurring.

A human eye cannot distinguish a change in intensity of upto 5 gray levels.

Size of the smallest square averaging mask which when applied on image will make the black square indistinguishable from the background for a human observer



Refer figures 3.34 (pg 156) & 3.33 (pg 155)

Non linear filter

Usually uses conditional operations (if, then..) & statistical (sorting pixel val in the neighbourhood etc)

→ Median filter → Max & min filter → Midpt filter

Refer figure 3.35 (pg 157) for the uses of median filter (salt & pepper noise)

Order statistic filters.

→ **Median**: Selects middle value of the distribution & is good for dist. with outliers
> Excellent at noise removal, w/o smoothing effects (impulse noise)
> Doesn't blur the edges (as much as simple averaging filters)

→ **Max & min**:

Max → $\hat{f}(x,y) = \max_{(s,t)\in S_{x,y}} \{g(s,t)\}$ Min → $\hat{f}(x,y) = \min_{(s,t)\in S_{x,y}} \{g(s,t)\}$

Max filter → good for pepper noise
Min " → " " salt "

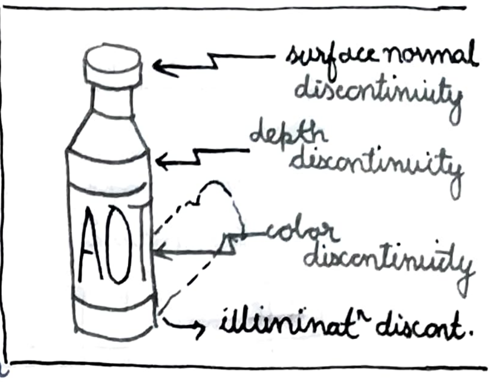
Image Sharpening

It's to enhance detail that has been blurred (perhaps due to noise or other defects, such as motion during image acquisition) or by enhancing the high frequency components

Geometric events → Surface orientat^n (boundary) discont, depth, color & texture discontinuities.

Non geometric events

Illumination changes secularities, shadows, inter-reflections

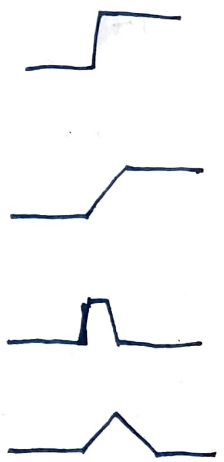


→ The objective of sharpening is to highlight the details in the image that has been, either in error or a natural effect of the method of image acquisition

→ It can be done by: ⟷ enhancing discont. in intensity

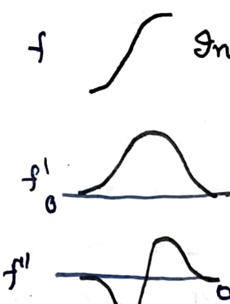
⟷ Edges represent borders between regions on an object or in a scene and has a significant intensity change in image.

Type of discontinuities in intensity



Derivatives of Intensity

Intensity discontinuity is detected by differentiating with neighbouring pixels.



Find maximas of derivatives of intensities

→ First derivative is non zero along the entire ramp, while second derivative is non zero only at the onset & end of ramp.

→ For a pt, response is much stronger for the 2nd derivative than the first.

→ For a step the 2nd derivative gives rise to a double response. This is used to detect edges

→ 2nd derivative is better suited than the first derivative for enhancing fine detail

First & second derivative

$$f(x+h) = f(x) + h\frac{\partial f}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 f}{\partial x^2} + \dots$$

$$\frac{\partial f}{\partial x} = \cancel{f(x+} \lim_{h\to 0} \frac{f(x+h)-f(x)}{h} \rightarrow \frac{\partial f}{\partial x} = f(x+1) - f(x) \text{ (for small } h \rightarrow \text{ step of 1)}$$

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

First derivative mask

$$\boxed{\frac{\partial f}{\partial x} = f(x+1) - f(x)} \rightarrow \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

→ estimate gradient; ↓ smoothing

$$\boxed{\frac{\partial f}{\partial y} = f(y+1) - f(y)} \quad \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

→ smoothing; ↓ estimate grad

Figure 3.36

2nd ~~First~~ derivative mask

approx. discrete fn mask

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

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

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

Derivative spatial mask

> Combined horizontal & vertical edge estimates are also used.

> A number of edge operators are described by a collection of masks

> Simplest : Robert's edge detector & inaccurate localizatn

1	0
0	-1

0	1
-1	0

→ Robert operator.

Refer to slides for an example on Robert op & First derivative operators
" " " " another " " 2nd order derivative operators

Laplacian derivative operator

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad \nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

→ Since sum of all weights is 0, the resulting signal has a 0 DC value.

→ The Laplacian derivative operator is used to highlight gray level discont. in an image & de-emphasizes regions with slowly varying gray levels.

→ Background features can be recovered while still preserving the sharpening effect of the Laplacian operatn by simply adding the original & the Laplacian images.

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{(if center coeff is -ve)} \\ f(x,y) + \nabla^2 f(x,y) & \text{(if center coeff is +ve)} \end{cases}$$

Refer Fig 3.37 in the Text book. For practical applicatns → check Fig 3.40 & 3.41

Unsharp masking & high boost filtering

Subtracting a blurred version of an image from the image itself

$$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$$

$$g(x,y) = \bar{f}(x,y) + k * g_{mask}(x,y); \quad k = 1$$

High boost filtering : $g(x,y) = \bar{f}(x,y) + k * g_{mask}(x,y)$, $k > 1$

$\boxed{09/09/20}$ – L11

Robert & Sobel operators.
Refer to fig. 3.41 & 3.42.

Prewitt operator

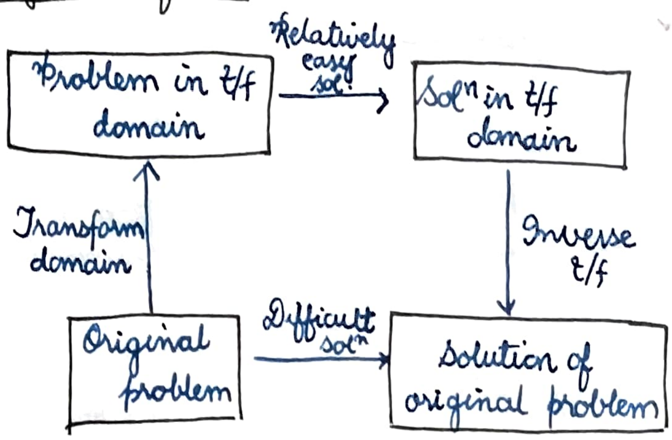
$$\text{smooth}\downarrow\ \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}\ \xrightarrow{}\ \text{gradient estimate}$$

$$\xrightarrow{\text{smoothing}}\ \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}\ \downarrow\text{gradient}$$

Combining spatial enhancement

→ Laplacian to highlight fine detail → Refer Fig 3.46
→ Gradient to enhance prominent edges
→ Smoothed version of the gradient image used to mask the Laplacian image.
→ Inc. the dynamic range of the gray levels by using a gray-level transformat^{n}s

Image transform



Need of image t/f

→ Better image processing
> Take into account long-range correlat^n in space.
> Conceptual insight on spatial freq. info.

→ Fast computat^n

→ Alternative representat^n
> Transformed data maybe compressed

→ Efficient storage & transformat^n
> Energy compact^n
> Pick a few representatives
> Store/send the contribut^n from representatives

Desired properties of transform

→ Reversible
→ Small no. of transform coeffs
→ Transform coeffs → uncorrelated
→ Transform : orthonormal
→ Low computational complexity of computing coeffs etc.

DFT

Advantage of working in freq. domain
→ Freq coeffs are segregated into low & high image freq.
→ Our visual system is less sensitive to distortion around edges
→ Transition associated with edge → masks our ability to perceive the noise

2D – FT

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(ux/M + vy/N)} \qquad \begin{matrix} u = 0,1,2 \dots M-1 \\ v = 0,1,2 \dots N-1 \end{matrix}$$

2D – IFT

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(ux/M + vy/N)} \qquad \begin{matrix} x = 0,1,2 \dots M-1 \\ y = 0,1,2 \dots N-1 \end{matrix}$$

u,v → Freq/Transform var. x,y → The spatial or image var.

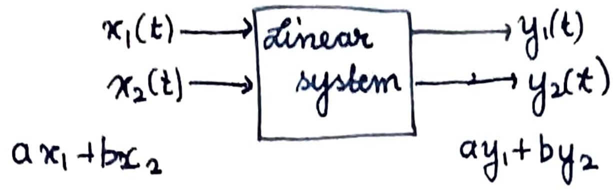
Fourier spectrum : $|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$

Phase : $\phi(u,v) = \arctan\left(\frac{I(u,v)}{R(u,v)}\right)$

Power spectrum : $P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$

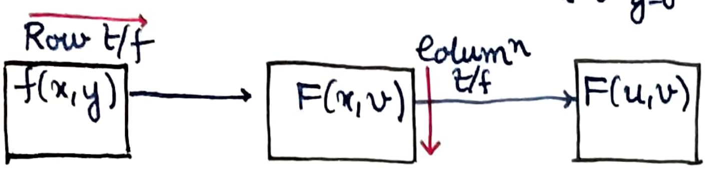
Properties of DFT

> **Linearity** :



FT is a linear image processing method.

> **Separability** : $$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi j(ux/M + vy/N)}$$



> **Rotation**

> **Expansion** : Expanding the original f image by a factor of n, filling the new values with zeros results in same DFT.

⟷ Functions : Sinc, rect, Gaussian, impulse

> **Averaging** : $$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi j(ux/M + vy/N)}$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{2\pi j(ux/M + vy/N)}$$

$$\bar{f}(x,y) = F(0,0)$$

$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$: Fourier spect.

$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$: Phase spect

Power spect : $P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$

DFT recap

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi j(ux/M + vy/N)}$$

So using $F_{uv} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} e^{-2\pi j(ux/M + vy/N)}$

$F_{u,v}$ & $F_{-u,-v}$ have similar frequencies but inverted shifts

Matlab: F= fft2(f) → has complex entries

D= log(1+abs(F)); reduce dynamic range of $|F(u,v)|$

fftshift(D); cyclically rotate the image s.t. $F(0,0)$ is @ center

> Visualizatⁿ of Fourier image using MATLAB.

```
% 30x30 blackimage
f= zeros(30,30);
f(5:24,13:17) = 1;
imshow(f);

% Calculate the fft
F=fft2(f);
F2=abs(F);
figure, imshow(F2,[])
```

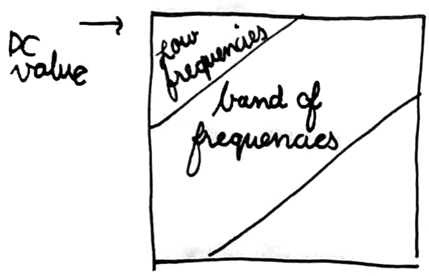
```
F= fft(f,256,256);
F2= abs(F);
figure, imshow(F2,[])
```

For better visualizatⁿ (adjusted as per human eye vision)

```
F2= abs(F2)
F2new= log(1+F2);
figure, imshow(F2,[])
```

Discrete Cosine transforms

DCT helps to separate the image into parts or spectral sub-band



Formula

$$r(x,y,u,v) = s(x,y,u,v)$$
$$= \alpha(u)\alpha(v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2M}\right]$$

where $\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{for } u=0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1,2,\ldots n-1 \end{cases}$

DCT implementation

$$F(u,v) = \frac{\alpha(u)\alpha(v)}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2M}\right)$$

DCT basis fⁿs

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2M}\right]$$ → Normalized values

> For most images, most of the energy → in low freq., upper left corner of DCT.
> Compression is achieved since: higher freq. are often small and in lower right corner, small enough to be neglected with little visible distortion

Decorelatⁿ: Low redundancy b/w neighbour pixels → uncorrelated coeffs which can be coded independently

Energy compactness: packing i/p data in as few coeffs as possible

Separability ÷ 2D DCT can be calc. ~~independen~~ in 2 steps by applyin' 1D formula in the lines & columns of an image.

Example in slides

Relatⁿ b/w DCT & FFT:

→ Only real part of FFT

→ Computationally simpler than FFT

→ Effective for multimedia compression.

DCT basis functions → Normalized values
→ DCT coefficient variance is less

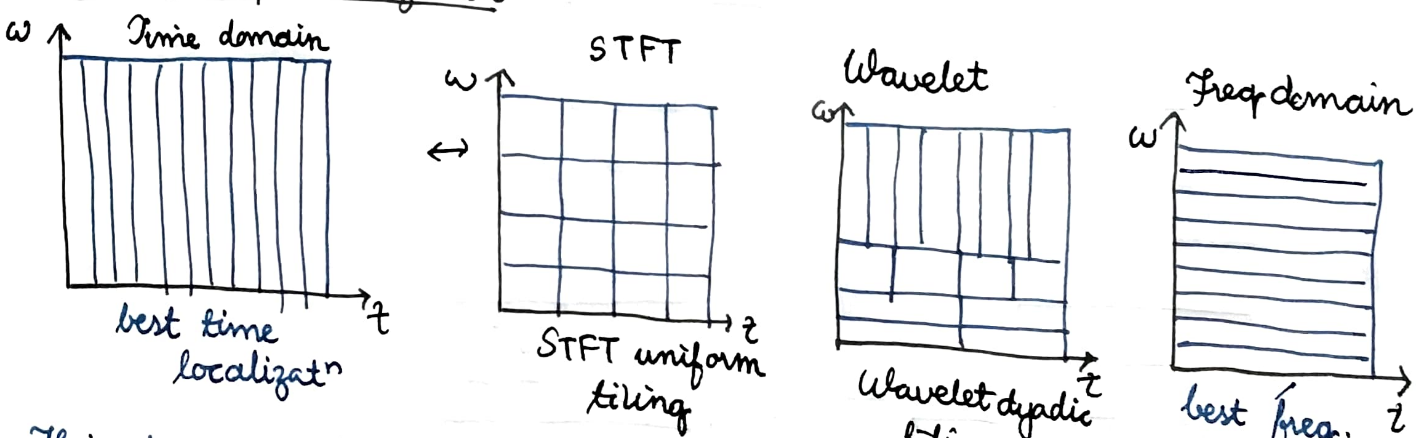
Ch7) Wavelets & Multi resolution processing.

Local histograms can vary from one part of the image to another making statistical modelling over the span of entire image is difficult or nearly impossible.

→ FT cannot tell where a particular feature rises
→ Short time FT : sliding window is used to find spectrogram → which gives info of both spatial & freq. domain. The length of the window limits the resolution in freq.

Wavelet T/f represents image in multi-resolut^ns.
Its eq. to a musical score, which reveals what notes to play & when to

Time - Freq Localization



→ Heisenberg uncertainty principle → bound on T-F product. $(S_0 W_0 \geq 1/4\pi)$
→ Wavelets provide flexibility & good time freq trade off.
> Low freq : wide window (because signal changes slowly)
High " : thin "
→ Flexibility can be achieved in wavelet domain by Multi resolut^n analy-sis.

History of wavelet transform
→ Slides & TB.

DWT
Fourier: basis f^n are sines & cosines ; Wavelet: set of wavelet f^ns

Wavelets

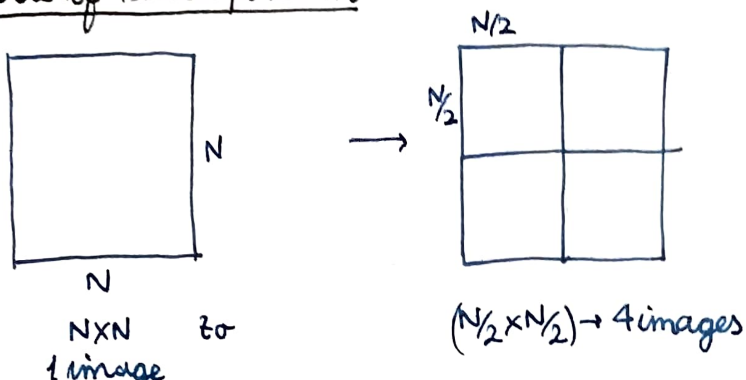
> Wavelet t/f performs a correlatⁿ analysis, therefore the o/p is expected to be maximal when the ~~sc~~ i/p signal resembles the mother wavelet

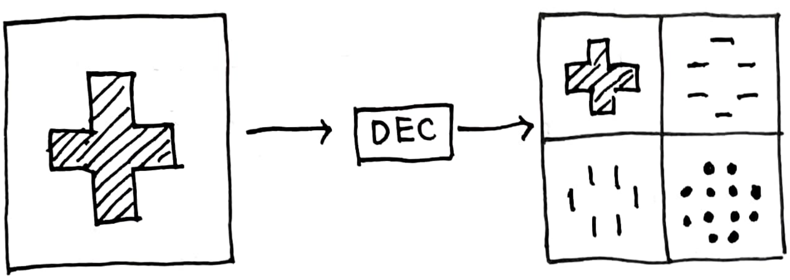
→ Decompose a signal into component parts
  > Fourier analysis: signal as a sum of sine & cosine fⁿs.
  > Signal becomes a set of wavelet coeffs. Coeffs represent features of a signal
  > Signal → completely constructed back from all coeffs
  > " → partially " " " some ".

> Small size objects or low contrast : normally examined at high resolutⁿs.
> Large " " " high " : a low res or coarse view is reqd.

Levels of decomposition:

MATLAB fⁿ:
DWT2(f)



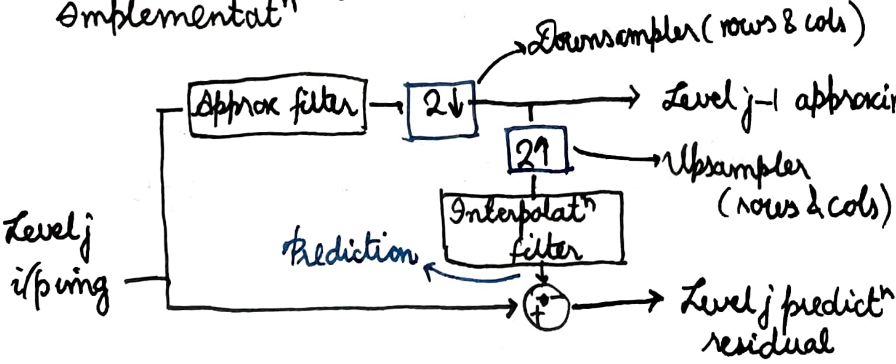


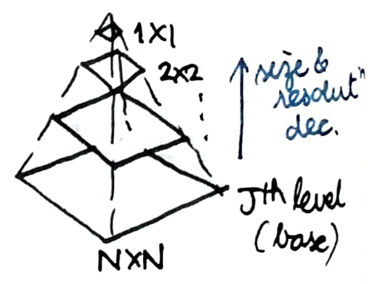
> Inherent multiresolution nature
> Image coding standard: JPEG-2000

Image Pyramid.

- A powerful structure to represent images @ more than one resolutⁿ.
- Dec resolutⁿ images arranged in the shape of a pyramid.

Implementatⁿ





DWT2(I, 'db2')
iDWT2(↓) → Decomposition & edge detection example.
Wavelet specs.

Freq. domain filtering

To filter in freq. domain →
- Compute $F(u,v)$, the DFT of the image.
- Multiply $F(u,v)$ by a filter f^n $H(u,v)$
- Compute inverse DFT of the result.

Convolution properties

Spatial component

$f(x,y) \to \boxed{h(x,y)} \to g(x,y)$

Freq. component

$F(u,v) \to \boxed{H(u,v)} \to G(u,v)$

$g(x,y) = f(x,y) * h(x,y)$

$G(u,v) = F(u,v) \cdot H(u,v)$

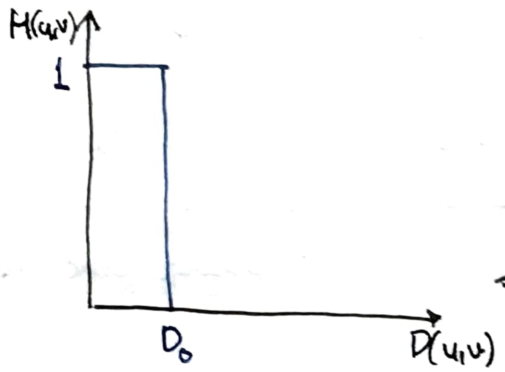
Types of filters

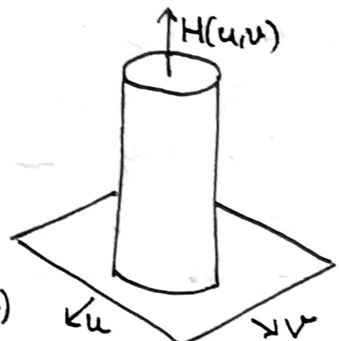
→ High pass → Low pass
→ Band pass → Band stop

Low pass filter

Allow low freq to pass, stop the high freq.

Ideal low-pass



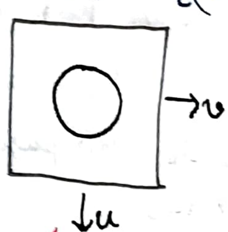


$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) \ge D_0 \end{cases}$$

where

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

≡



Example: Design ILPF (Ideal LPF) of size 4x4 & centre (2,2) & cutoff radius of 1.9.

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

$$D(k,l) = \begin{array}{c|cccc} & 0 & 1 & 2 & \\ \hline 0 & 2.8284 & 2.2361 & 2.0000 & 2.2361 \\ 1 & 2.2361 & 1.4142 & 1.0000 & 1.4142 \\ 2 & 2.0000 & 1.0000 & 0 & 1.0000 \\ & 2.2361 & 1.4142 & 1.0000 & 1.4142 \end{array}$$

$$H(k,l) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

2D DFT of averaging spatial filter →
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \to F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

$u = 0, 1 \ldots M-1$; $v = 0, 1, 2 \ldots N-1$

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

$F(u,v) = F(0,0) = 1$ → DC value

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

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

Ideal LPF

$$P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{Q-1} P(u,v) \rightarrow \alpha = 100 \left[\sum_u \sum_v P(u,v)/P_T \right] \rightarrow$$ circle enclosing specified amt of total img power.

Fig 4.41 & Fig 4.42 from the text

Drawback: Ideal LPF

→ Cutoff freq is sharp & hence cannot be realised by electronic components, although can be simulated in a computer.

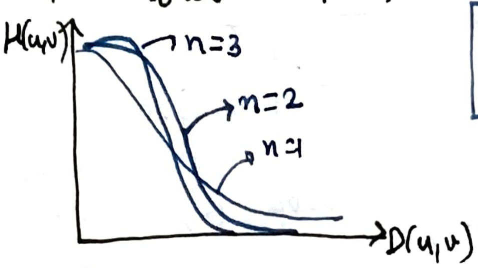
$$H(k,l) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Effects of using such non physical filters on a digital image causes ringing effect

→ Most of the sharp details in 13% power removed by the filter

blob: binary large objects?

Butterworth low pass filter

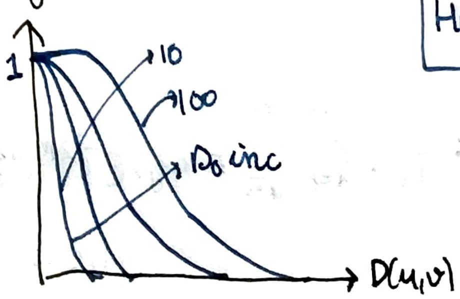


$$\boxed{H(u,v) = \frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}}$$

No sharp discontinuity

Ringing with 'n'
n=1 → no ringing

Gaussian LPF



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

Table 4.4 LPFs.
Fig 4.19 to Fig 4.24

Ideal HPF

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

ILPF → 4×4, centre (2,2) rad = 1.9

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

$$D(k,l) = \begin{bmatrix} \sqrt{8} & \sqrt{5} & \sqrt{4} & \sqrt{5} \\ \sqrt{5} & \sqrt{2} & \sqrt{1} & \sqrt{2} \\ \sqrt{4} & \sqrt{1} & 0 & \sqrt{1} \\ \sqrt{5} & \sqrt{2} & \sqrt{1} & \sqrt{2} \end{bmatrix} \rightarrow H_h(k,l) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = I - H_{ll}(k,l)$$

> Point processing
> Spatial processing
> Freq processing.

Image Restoration

> Identify the degradatn process & attempt to reverse it.

Enhancement

→ Subjective process

→ Priori ~~imag~~ knowledge abt the img is not a must

→ Procedures are heuristic & take advantage of HVS's psychophysical aspect

Ex.) Contrast stretching

Restoration

→ More of an objective process

→ Images are degraded

→ Images recovered by using the knowledge abt the degradation process

Ex) Deblurring

Sources of degradatn.

→ Ambient conditions

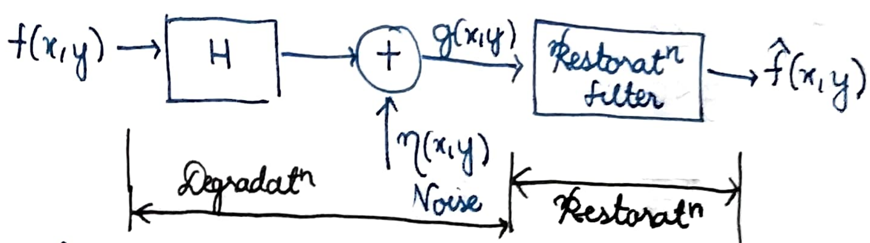
→ Atm. noise

→ CCD & stepper motor noise

→ Interference while transmission

→ Film grain noise

→ Network transmission errors.



→ Image degradatn model.

$$g(x,y) = f(x,y) * h(x,y) + \eta(x,y)$$

Image restoration : spatial or freq domain

1) Spatial → noise is additive

2) Freq domain → image blur

Noise: indep of spatial coord.   components

↳ Uncorrelated w.r.t. the image (if no correl^n b/w pixel values & noise components)

Additive noise:

$$g(x,y) = f(x,y) + \eta(x,y) \qquad h(x,y) = \delta(x,y)$$

$g(x,y)$ ↓ noisy pixel; $\eta(x,y)$ ↓ noise

Noise models

→ Gaussian (common)   → Rayleigh   → Gamma (also called Erlang)   → Exp.

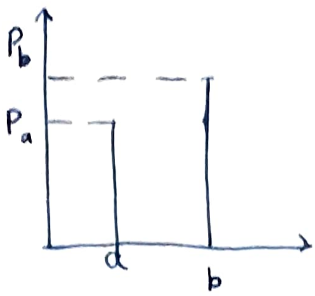
→ Uniform   → Impulse. (Salt & pepper noise)

Noises: defined as random var. following a PDF as a spatial descriptor

→ The mathematical model for Rayleigh noise

$$P(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & ; \text{ for } z \geq a \\ 0 & ; \text{ for } z < a \end{cases} \qquad \bar{z} = \frac{b}{a}, \quad \sigma^2 = \frac{b}{a^2}$$

→ Impulse (S & P noise)

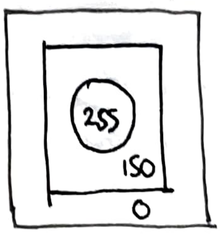


$$P(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{o/w.} \end{cases}$$

For 8bit gray image

$a = 0$

$b = 255$



→ Test Image

Gaussian → $p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(z-\mu)^2/2\sigma^2}$; $z \geq a$

0 $z < a$

$$\text{Gamma} \rightarrow \quad p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & ; z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$\text{Exp} \rightarrow \quad p(z) = \begin{cases} a e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$\text{Uniform noise} \rightarrow \quad P(z) = \begin{cases} \frac{1}{b-a} & ; a \leq z \leq b \\ 0 & ; \text{o/w} \end{cases}$$

22/09/20 - L16

Noise will be assumed to be:
→ indep of spatial coordinates → uncorrelated w.r.t. the image

Noise models: $\eta(x,y)$
→ Gaussian → Rayleigh → Gamma → Exponential (Erlang)
→ Uniform → Impulse.

Estimat^n of Noise Parameters

Case 1: Imaging system is available : capture images of flat environment

Case 2: Imaging system is available: Capture multiple images

Case 3: Noisy images available : > Take a strip from const. area
> Draw histogram & observe it
> Measure the mean & variance.

Rayleigh Noise

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & ; z \geq a \\ 0 & ; z < a \end{cases} \qquad \bar{z} = \frac{1}{a} \quad , \sigma^2 = \frac{1}{a^2}$$

Impulse Noise

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & o/w \end{cases}$$

Histogram is an estimate of noise PDF

$$\mu = \sum_{z_i \in S} z_i \, p(z_i) \qquad \sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)$$

Arithmetic mean filter : $\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$ $g(s,t) \rightarrow$ noisy image, $m \times n \rightarrow$ size of filter.

Implemented as simple smoothing filter, blurs the image.

Geometric mean filter: $\hat{f}(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{1/mn}$

Achieves blurring like smoothing (avg-filter : AM) but loses less image detail

Harmonic mean filter: $\hat{f}(x,y) = \dfrac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$ → Works well for salt noise but not for pepper noise. Does well for Gaussian noise as well.

Contraharmonic mean filter : $\hat{f}(x,y) = \dfrac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q}}$

$Q \rightarrow$ order of the filter
$Q > 0 \rightarrow$ eliminate pepper noise
$Q < 0 \rightarrow$ " salt "
Can't eliminate both simults.

Look up examples in the TB for all these filters.

Order statistics filter: Based on ordering / ranking of pixels that make up the neighbourhood defined by filter support.

→ Median filter

$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\}$

→ Excellent noise removal, no smoothing filters.

→ Very good for S&P noise

→ Max & min filter

$\hat{f}(x,y) = \max\{g(s,t)\}$

$\hat{f}(x,y) = \min\{g(s,t)\}$

Max → S Noise

Min → P noise

→ Midpt filter

Midpt b/w max & min values of intensity values calc.

→ Good for Gaussian & Noisy

→ Alpha trimmed mean filter

$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$

Applicat^n in TB.

Estimating the degradat^n f^n

> Estimat^n by image observat^n
> " " experiment
> " " modelling

$g(x,y) = f(x,y) * h(x,y) + \eta(x,y)$

$G(u,v) = F(u,v)H(u,v) + N(u,v)$

Observat^n: degradat^n f^n can be estimated by → $H(u,v) \approx H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$

Expt: Obtain an impulse response of the degradat^n using the system setting
A linear space invariant system is characterized by its impulse response.

$H(u,v) = \frac{G(u,v)}{A}$

Modelling: Hufnagel & Stanley (1964) → $H(u,v) = e^{-k(u^2+v^2)^{5/6}}$
(due to atm turbulence)

k → determined by expts

Estimatn by modelling

Motion blurring : camera vel $\rightarrow x_0(t)$ & $y_0(t)$

Blurred image obtained by : $g(x,y) = \int_0^T f(x+x_0(t), y+y_0(t)).dt$ T → exposure time

$g(x,y) \leftrightarrow G(u,v)$

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\, e^{-j2\pi(ux+vy)}.dx\,dy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f(x+x_0(t), y+y_0(t))\,dt\right] e^{-j2\pi(ux+vy)}.dxdy$$

$$= \int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x+x_0(t), y+y_0(t))\, e^{-j2\pi(ux+vy)}.dxdy\right]dt = F(u,v)H(u,v)$$

$$H(u,v) = \frac{1}{T}\int_0^T e^{-j2\pi(ux_0(t)+vy_0(t))}.dt$$

Examples in slides & TB

Motion Blur mask coeffs.

X		X		
	X	X		
X	X	X	X	X
		X	X	
		X		X

— X → Horizontal
— X → Vertical.
— X → Diagonal

Blur mask coeff typical examples in slides.

Image Compression

→ Reduce redundancy of the img & store or transmit in efficient form.

Amt of data reqd to represent 2hr long std def TV movie using 720×480×24 bit pixel arrays.

$$30\,\frac{\text{frames}}{\text{sec}} \times (720\times 480)\,\frac{\text{pixels}}{\text{frame}} \times 3\,\frac{\text{bytes}}{\text{pixel}} = 31{,}104{,}000 \text{ bytes/sec}$$

$$31{,}104{,}000\,\frac{\text{bytes}}{\text{sec}} \times 60^2\,\frac{\text{sec}}{\text{hr}} \times 2\text{ hrs} \approx 2.24\times 10^{11} \text{ bytes} = 224 \text{ GB !!}$$

Compression ratio : $C_R = \frac{n_1}{n_2}$ $R_D = 1 - \frac{1}{C_R} = 1 - \frac{n_2}{n_1}$

$n_1 = n_2 \rightarrow C_R = 1$, $R_D = 0$ no redundant info

$n_2 \ll n_1 \rightarrow C_R \rightarrow \infty$, $R_D \rightarrow 1$ highly redundant data : significant compression

$n_1 \ll n_2 \rightarrow C_R \rightarrow 0$, $R_D \rightarrow -\infty$ (undesirable) → data expansion.

→ Coding redundancy → Interpixel redundancy → Psychovisual redundancy.

Example for coding redundancy

Gray val.	0	1/7	2/7	3/7	4/7	5/7	6/7	1
P(r)	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
codes	000	001	010	011	100	101	110	111
Var len.	11	01	10	001	0001	00001	000001	000000

Avg code length → 3 bits/pixel

Memory reqd → $512 \times 512 \times 3 = 7,86,432$ bits

Var len code : $L_{avg} = \sum_i p_i l_i = 2.7$ bits/pixel

Memory reqd → $512 \times 512 \times 2.7 = 7,07,788.8$ bits

$C_R = \frac{n_1}{n_2} \Rightarrow R_D = 1 - \frac{1}{C_R}$

Image Compression

Interpixel redundancy → > Parts of an image are highly correlated
> Pixel intensity can be predicted from its neighbour (Autocorrelation)

(RLE) **Run length encoding** → Runs of data are stored as a single data value & count rather than the original run. (Run → seq. in which 1 data val occurs in many consecutive data elements)

Example > $\underbrace{00000}_{5}$ $\underbrace{1111}_{4}$ $\underbrace{000000}_{6}$ $\underbrace{111111}_{6}$ Applicat^n^ - binary image, fax etc

(0,5) (1,4) (0,6) (1,6)

Code words → $\boxed{0101}$ $\boxed{1011}$ $\boxed{0110}$ $\boxed{1100}$

Q> 343 × 1024 image → binary
No. of runs given → 12166
Non-compressed → 343 × 1024 × 1 = 351,232
Raster scanning → 
RLE → 12166 × ~~10~~ 11 = ~~12,16~~ 133,826
2D info → 1D.

Line 100: (1,63) (0,87) (1,37) (0,5) (1,4) (0,556) (1,62) (0,210)
10 bit + 1 bit

Image isn't completely binary → check slide. Therefore we select a threshold value to demarcate into black & white. Threshold value → 125

$C_R = \frac{n_1}{n_2} = 2.63$, $R_D = 1 - \frac{1}{C_R} = 0.62$

Psychovisual redundancy → Non linear response of the eye to all visual info.

8bit grayscale → 4 bit grayscale (compression)

256 GS Val → 16 GS Val.
(False contouring)

4 bit IGS
(No false contouring)

Improved Grayscale Quant.

1> Add 4-LSB to previous value of sum to 8 bit current pixel. If 4MSB is 1111, then ~~the~~ add 0000 instead, keep the result in sum.

2> Keep only the 4-MSB of sum for IGS code.

Example

Pixel	Graylevel.
i-1	N/A
i	0110 1100
i+1	1000 1011
i+2	1000 0111
i+3	1111 0100

Pixel	Gray level	Sum	IGS Code
i-1	N/A	0000 0000	N/A
i	0110 1100	0110 1100	0110
i+1	1000 1011	1001 0111	1001
i+2	1000 0111	1000 1110	1000
i+3	1111 0100	1111 0100	1111

Measuring image information:

The entropy H(S) represents a fundamental limit on the avg. no. of bits or code length L per source symbol, necessary to represent a discrete memoryless source (DMS).

$$\boxed{L_{av} \geq H(s)}$$

This measure gives us with a theoretical min. for avg. no. of bits per pixel that could be used to code the image.

An info source cannot be determined if $L_{av} < H(s)$

19~~8~~48: noiseless coding (Shannon's first theorem)

$$I(E) = \log_2\left(\frac{1}{P(E)}\right) = -\log_2 P(E) \text{ bits}$$

I → Information in the event or self information of E

→ $P(E) = 1$, $I(E) = 0$ bit — Predictable signal conveys no info.

→ $P(E) = 1/2$, $I(E) = 1$ bit

→ $\underline{P(E) = 0}$, $I(E) =$ ~~infinite~~ 0

Source alphabet: $A\{a_1, a_2, \ldots a_J\}$; $\sum_{j=1}^{J} P(a_j) = 1$

$$z = [P(a_1), P(a_2), \ldots P(a_J)]^T$$

The avg info per source o/p → $\boxed{H(z) = -\sum_{j=1}^{J} P(a_j)\log_2(P(a_j))}$

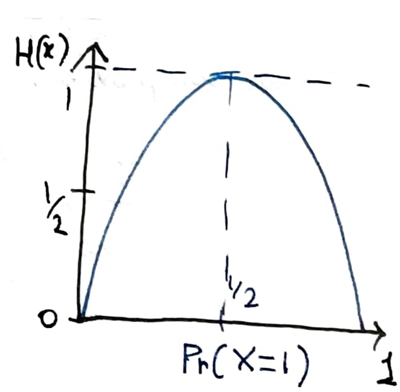
Example: Entropy

A binary image has equiprobable black & white pixels. Find the info content of bf. b & w pixels

$$P(b) = P(w) = \frac{1}{2}$$

$$I(E) = -\log_2(P(E)) = +1 \text{ bit each}$$

$$H(z) = -\sum_{j=1}^{J} P(a_j)\log_2(P(a_j)) = \frac{2}{2} = 1 \text{ bit}$$



If $P(w) = 0.95$ → $H(z) =$ ~~0.45~~ 0.2863 bits/pixel

$P(a_1) = 0.95$

$P(a_2) = 0.05$

Example:

21 → 00
95 → 01
169 → 10
243 → 11

H = 2 bits/pixel

→ optimal code for that image

Example

r_k	$P_r(r_k)$
$r_{87}=87$	0.25
$r_{128}=128$	0.47
$r_{186}=186$	0.25
$r_{255}=255$	0.03

$$H = -\sum_i p(a_i)\log_2 p(a_i)$$
$$= 1.6614 \text{ bits/pixel}$$

Example 8.10>

$p(a_{21}) = \frac{12}{32}$ $\quad p(a_{95}) = \frac{4}{32}$ $\quad p(a_{169}) = \frac{4}{32}$ $\quad p(a_{243}) = \frac{12}{32}$

$H = -\sum_i \qquad = 1.81$ bits/pixel

Example 8.11> Differential coding scheme.

$H = 1.41$ bits/pixel

Fidelity criteria:

Objective Fidelity ~~cri~~ criterion

RMSE

~~MS-SNR~~ PSNR

Subjective Fidelity criterion

→ Human rating

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

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

RMSE

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

MS-SNR

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

Image compression standards.

$$\underbrace{\underbrace{\text{Binary} \quad \text{Cont. tone}}_{\text{Still img}} \quad \text{Video}}_{\text{Image comp. stds, formats \& containers}}$$

Basic Compression methods.

> Huffman Coding $\nearrow$ 1952 → Applicat^n examples → JPEG, MPEG, MP3

→ Source prob model should be known.

→ Its systematic method to find codes of source events

→ Codes need not be unique.

Procedure →

> Source symbols listed in descending order

> Merge the two least probable outputs into one as composite code whose prob. is sum of the two corresponding probs.

> If the number of remaining o/p is 2, then go to next step otherwise go to step 2.

Example: $\underbrace{A = \{a_1, a_2 \dots a_6\}}_{\text{6 gray shades in img.}}$ $P = \{0.1, 0.4, 0.06, 0.1, 0.04, 0.3\}$

My method

a_1	0.1
a_2	0.4
a_3	0.06
a_4	0.1
a_5	0.04
a_6	0.3

→

a_2 0.4	1	0.4	1	0.4	1	0.6	0
a_6 0.3	00	0.3	00	0.3	00	0.4	1
a_1 0.1	0100	0.2	010	0.3	01		
a_4 0.1	0101	0.1	011				
a_3 0.06	0110						
a_5 0.04	0111						

$L = 2.2$

Sir's example

If composite code has ~~least~~ same prob. & then it has to be in the lower pref.

a_5 0.04	a_{53} 0.1	~~a_4 0.1~~ a_4 0.1	a_{5314} 0.3	0.6 a_{53146}
a_3 0.06	a_1 0.1	a_{531} 0.2	a_6 0.3	0.4 a_2
a_4 0.1	a_4 0.1	a_6 0.3	a_2 0.4	
a_1 0.1	a_6 0.3	a_2 0.4		
a_6 0.3	a_2 0.2			
a_2 0.4				

Slide

0.4	1
0.3	00
0.1	011
0.1	0100
0.06	01010
0.04	01011

$L = 2.2$

011, 0100 → ~~Sed~~ same code but diff notat^n.

Source extension: $\{a_1, a_2\} \longrightarrow \{a_1a_1, a_1a_2, a_2a_1, a_2a_2\}$

Source extension improves efficiency.

	1st extension	P	I	l	Code	Code length
a_1	α_1	$2/3$	0.59	1	0	1
a_2	α_2	$1/3$	1.58	2	1	1

2nd extension

		P	I	l	Code	Code length
a_1a_1	α_1	$4/9$	1.17	2	0	1
a_1a_2	α_2	$2/9$	2.17	3	10	2
a_2a_1	α_3	$2/9$	2.17	3	110	3
a_2a_2	α_4	$1/9$	3.17	4	111	3

$H = 0.918$

$L_{av} = 1$

$\eta = 0.918$

$H = 1.83$

$L_{av} = 1.89$

$\eta = 0.97$

→ Drawbacks:

- Poor efficiency for skewed prob. dist.
- Prob. dist. of intensities must be known.
- Need to store code words as lookup tables.
- Efficiency is improved by block code
- Hard to adapt to changing stats

Arithmetic coding

→ The entire sequence of pixel intensities are assigned a single arithmetic code word in the form of a number in an interval of a real number b/w 0 & 1.

→ String / sequence coding

→ Example: sequence of intensities $\{a_1; a_2; a_3; a_4; a_5\} \rightarrow a_1 a_1 a_4 a_4 a_4 \ldots$

> Assume all pixels are concatenat'n of gray shades from Gray level values.

> Fn maps random intensity ~~values~~ variables & sequences of random intensity variables to CDist. fn.

> This is done by a set of interval ranges determined by the prob.

> A unique identifier ~~is~~ or tag is generated for the intensity sequence to be coded.

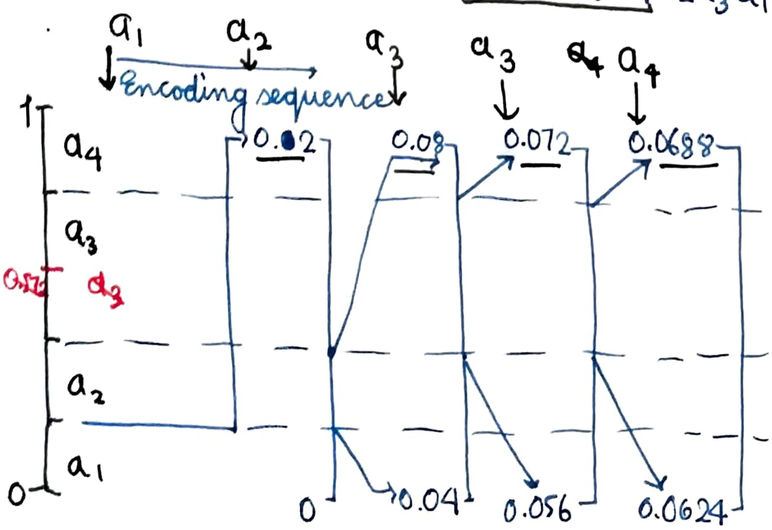
> The tag corresponding to binary fract^n. → no. are ~~b/w~~ in [0,1) & finite

> The number is unique & in order to map this, we need a f^n that'll map seq. of intensities in the unit interval

Example: 10×10 image

Intensities	Prob.	Sub interval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1)

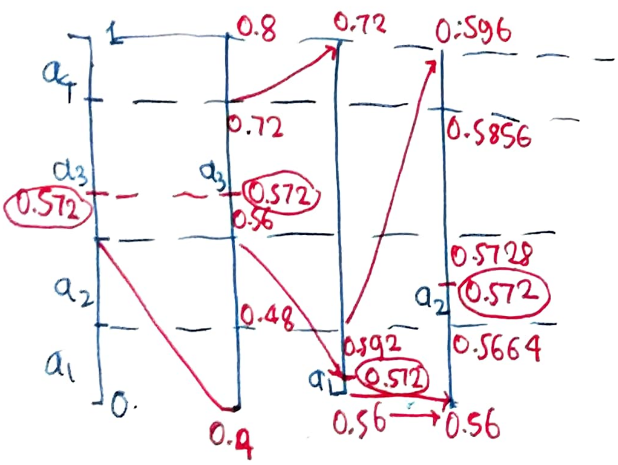
Code for 1st row: $a_1 a_2 a_3 a_3 a_4 a_2 a_3 a_1 a_4 a_2$



[0.0688, 0.06752]

Decode the sequence 0.572

$a_1 \rightarrow 0.2$
$a_2 \rightarrow 0.2$
$a_3 \rightarrow 0.4$
$a_4 \rightarrow 0.2$



$a_3\ a_3\ a_1 a_2 \ldots$ & so on.

Difficulty → Shrinking the interval requires very high precision

Lempkel Ziv Coding (LZ)

> Huffman & Arithmetic → sequence of indep symbols generated
> Src coding can be compressed more if we use correlatⁿ or structural info. of the data
> LZ technique : used for structural info of data
> Based on 'code book' → build a list of commonly occuring patterns & encode these patterns by transmitting the index to the book.
> It doesn't require source statistics.
> Assign ~~so~~ fixed length code words to var. length sequences of source symbols.

R1 39 39 126 126 24 24 150 150

same pattern as R1 { ″ ″ ″

Let's consider the 4x4 part of this image: 39 39 126 126

Convert 2D data to 1D by Raster type scanning. same as R1 { — — —

{39, 39, 126, 126, 39, 39, 126, 126 . . . . }

S1 → Initialize a dictionary by noting all possible gray values (0-255)
S2 → I/p current pixel
S3 → If 'current pixel + previous' pixel' form ~~one~~ one of existing dictionary entries
Then
2.1 → Move to next pixel & go to S1.
2.2 → Else : O/p the dictionary locatⁿ of the currently recognized sequence in ~~2x~~ 2.2 with the current pixel.
2.3 → Create a new dictionary entry by appending ~~entry by appending~~ the currently recognized sequence in 2.2 with the current pixel.
2.4 → Move to next pixel & repeat S1

Parsing patterns in 1-D data

(39, 39) (39, 126), (126, 126) (126, 39), (39, 39, 126) (126, 126, 39), (39, 39, 126, 126)

Locatⁿ	Dictionary Entry
0	0
1	1
⋮	⋮
255	255
256	39-39
257	39-126
258	126-126
259	126-39
⋮	⋮

i/p pixel → 39 39 126 126 39 39 126 126 39 39 126 126

Currently recognized seq.	Pixel being processed	Encoded output
—	39	—
39	39	39
39	126	39
126	126	126
126	39	126
39	39	
39-39	126	256
126	126	
126-126	39	258
39	39	
39-39	126	
39-39-126	126	260
126	39	
126-39	39	259
39	126	
39-126	126	257
126	—	126

9 bits used in encoding

39, 39, 126, 126, 256, 258, 260, 259, 257, 126

08/10/20 - L23

Bit plane compression

Binary bit plane compression method.

Original gray level: $11001 = 2^4 \times 1 + 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 1$

Bit plane level	Coeff
4	1
3	1
2	0
1	0
0	1

Example of binary compression → run length coding

Graycoded bit-planes: $g_i = a_i \otimes a_{i+1}$ for $0 \le i \le 6$ & $g_7 = a_7$

$\otimes \to$ XOR

Less transitions in gray coded bit planes ⇒ more efficient for coding.

JPEG: Joint photographic experts group: → ISO/IEC JTC1 / SC 29/WG 10

JPEG old: based on pred. rules

JPEG base line: lossy coding of cont. tone still images → based on DCT

JPEG LS → predictive & entropy coding

JPEG 200 → based on DWT

JPEG old

→ Lossless as well as lossy.
→ 29 distinct coding schemes.
→ 7 predictive schemes

→ Three based on 1-D pred.
→ Four based on 2-D pred.

Predictive schemes:

1D → 1. $\hat{I}(i,j) = I(i-1,j)$ 2. $\hat{I}(i,j) = I(i,j-1)$ 3. $\hat{I}(i,j) = I(i-1,j-1)$

2D → 4. $\hat{I}(i,j) = I(i,j-1) + I(i-1,j) - I(i-1,j-1)$

5. $\hat{I}(i,j) = I(i,j-1) + \dfrac{[I(i-1,j) - I(i-1,j-1)]}{2}$

6. $\hat{I}(i,j) = I(i-1,j) + \dfrac{[I(i,j-1) - I(i-1,j-1)]}{2}$

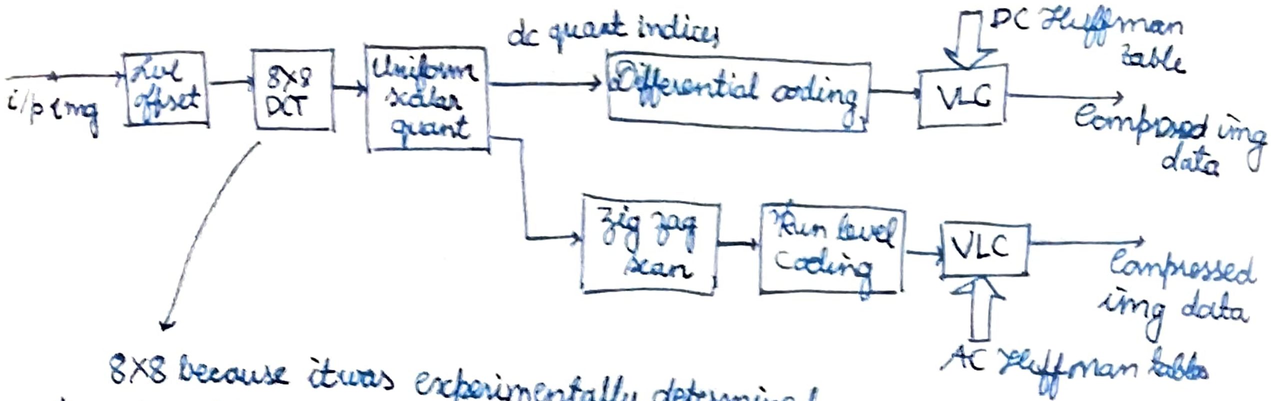
7. $\hat{I}(i,j) = \dfrac{[I(i,j-1) + I(i-1,j)]}{2}$

→ In offline process, all predictors are tested.
→ 3 bit headers are added for predictor info
→ The residual images are coded by adaptive arithmetic coding.

JPEG based on transform →

A general compression technique independent of

> Image resolutn
> Color system
> Image & pixel aspect ratio
> Image complexity



8X8 because it was experimentally determined to give optimal results

HEVC → High efficiency video coder (64X64 DCT)

Image processing & JPEG

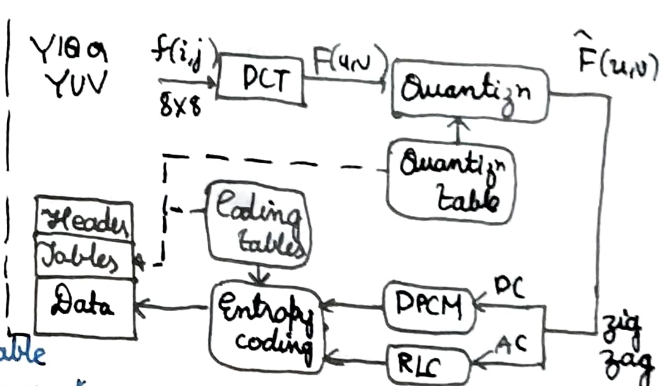
Block Transform coding

JPEG is effective because of the following three observat'ns.

1. Image data usually changes slowly across an image, esp within an 8x8 block
2. Therefore images contain much redundancy.
3. Experiments indicate that humans are not very sensitive to high freq data images. Hence we can remove much of it using transform coding.

Example 8.17

JPEG Coder



$$q_{val} = round\left(\frac{dct2}{q_{mt}}\right)$$

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2

DC value.

0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

DC value of previous DCT block: -17

$-26-(-17) = -9$ → Check the table : DC table category 4

Base code: 101 → length of the code is 7, remaining 4 bits will be LSBs of -9 which is 0110

8 → 1000

9 → 1001 → 2's compl. → 0110

Final code → 1010110

-3 → 11 → 00 ; From table 01 00

Finally we get : → $C_R = \frac{(8\times8)\times8}{92} = 5.6$ $\quad R_D = 1-\frac{1}{5.6} = 0.82$

→ Revision of JPEG coding ; JPEG 2000: Read slides

Disadvantages of DCT Compression:

- DCT based JPEG was blocks of image, there's correlation across the blocks
- Block boundaries are noticeable in some cases.
- Blocking artifacts at high compression
- Block overlap
- Computationally expensive

JPEG DCT compression artifacts:

→ One common artefact of block based t/f coding is the "blocking" effect or "grid noise".
The blocking effect is a result of the approx. of the coeff during quantiz^n
↳ Becomes more pronounced as the quantizat^n steps are coarsened ~~do~~ to reduce the data rate.
Its a particularly problematic issue along the edges or color boundaries in the image or in areas of image that are relatively uniform in colour.

Ringing artifacts:

→ Ringing artifacts are the result of quantizat^n error of high spatial freq.
→ 1D case: If we zero-out the higher freq. coeffs, some fuzziness is observed in the recovered signal.
→ Most likely to occur along strong edges in an image & is common in text because of the prevalence of strong edges. Ringing artifacts would manifest as fuzzy grey lines in vicinity of an edge & detectable only if surrounding pixels have a relatively higher luminance

Wavelet & encoder & decoder

I/p → Wavelet transform → Quantizer → Symbol encoder → Compressed image

Compressed image → Symbol decoder → Inverse wavelet transform → Decompressed image

Wavelet compression

→ "Modern" lossy wavelet coding exploits multi-resolut^n & self-similar nature of wavelet decomposition
- Energy is compacted into a small number of coeff.
- Significant coeff. tend to cluster at the same spatial locat^n in each frequency subband.

→ Two set of info. to code:
- → Where are the significant coeffs?
- What values are significant coeffs?

→ Parent children relat^n among coeff.
- → Each parent coeff @ level k spatially correlates with 4 coeff at k-1 of same orient^n.
- A coeff at lowest band correlates with coeff.

→ Coding significance map via zero Tree
>> Encoding only ~~needs~~ the high energy coeffs, no need to send info locatⁿ; large overhead
>> Encode insignificance map with zero trees
→ successive approx quantizatⁿ
>> Send MSB first & then refine coeff values
>> Embedded nature of coded bit-stream : get hi-fi image by adding extra refining bits

Image Morphology Processing

Morphological processing is constructed with operatⁿ on sets of pixels.

Binary morphology is used to extract image components that are useful in the representatⁿ & descriptⁿ of region shape, such as:

1. Boundary extractⁿ
2. Skeletons
3. Convex Hull
4. Morphological filtering
5. Thinning
6. Pruning

Morphological HP Img Processing

→ A collectⁿ of non linear operatⁿ related to shape or morphology features in an image.
→ Probing an img with a shape or template called structuring element.

→ Based on algebraic set theory
→ Sets in mathematical morphology ≡ objects in image
→ A tool for extracting image components that are useful in the representⁿ.
→ Morphological operator: defined by structuring element & the applied set operator
→ If 2 sets of elements match the condⁿ defined by the set operator, the pixel neighbouring the origin of the structuring element is set to a pre defined value (0 or 1 for binary images)

Structure element : small binary image Refer Fig 9.2 TB

A: fit
C: neither hit nor fit
B: hit

Set theory in pictures

Let A be a set in Z^2, if $a = (a_1, a_2)$ is an element of A then.
If $a \notin A$
> Null or empty set

Subset: $A \subset B$
Disjoint: $A \cap B = 0$
Union: $A \cup B$
Complement: $\bar{A}$
Intersectⁿ: $A \cap B$
Difference: $A - \cancel{B}(A \cap B)$
Translatⁿ: $(A)_z = \{c \mid c = a + z, \text{ for } a \in A\}$
Reflectⁿ: $\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$

Dilation

→ To compute dilation, we consider each ~~element~~ pixel in the background in the i/p image.
→ For each bckgnd pixel, we superimpose the structuring elements on top of the i/p image so that the origin of the structuring element coincides with i/p pixel posⁿ
> $A \oplus B = \{z \mid z = x + y, x \in A, y \in B\}$ → dilatⁿ of ~~A&+~~ A by B.
where B is the structuring element
> The dilatⁿ of A by B is the set of all displacements, z, such that $\hat{B}$ & A overlap atleast on element.

IF atleast one pixel in the structuring element coincides with a foreground pixel in the image ~~undew~~ underneath, then the i/p pixel is set to the foreground value. If all the corresp. pixels in the image are bckgnd, however, the i/p pixel is left at the background value.

Bridging gaps:
→ The max length of the breaks is known to be 2 pixels.
→ A simple structuring element that can be used for repairing gaps → ref to fig 9.5b (TB)

Erosion

Erosion of A by B, denoted by $A \ominus B$; $A \ominus B = \{z \mid z - B \subseteq A\}$
where B is referred to as the structuring element
Erosion of A by B : set of all. pts z s.t. B, translated by z is contained in A
Fig 9.3 (TB)
Fig 9.5 (TB)

Opening operatⁿs:

i/p image is subjected to both dilatⁿ & erosion : Opening operatⁿ opens small gaps b/w touching objects in an image.

$A \circ B = (A \ominus B) \oplus B$ → A opened using B → generally smoothes the contour of an object.

Closing operat^n: $A \bullet B = (A \oplus B) \ominus B$ → A closed by B.
See examples in TB.

Image Segmentat^n

Thresholding is subjective!

Goal is to find individual objects in an image & we stop once the object of interest is isolated

> Segmentat^n → Partition into disjoint regions
> Various methods find interiors of the border of an object
> Border of a regn is found is using edge detect^n
> The interior of the region is determined by distinct properties of pixels comprising the regn
> Thresholding, watershed, region-based methods.

Partitioning into a set of N connected regns $R(n) = 0, 1 \ldots N-1$

$$P(R) = \begin{cases} 1 & \text{if } f(R) \in H \\ 0 & \text{, o/w} \end{cases}$$ → determining the interior part of an img

1. Sum of all regns → equal to image
2. A pixel of img belongs to only one of the regns
3. Properties of one regn is same for all pixels
4. Properties of adjacent regions must be different

See Fig 10.1 for reference.

Applicat^{n}s: Count no. of objects, geometric property measurement of objects in image, study properties of individual objects (intensity, texture etc)

Two basic properties of pixel intensities

Discontinuities
> Pt detect^n
> Line detect^n
> Edge "
> Object boundary

Similarities
> Thresholding
> Regn growing
> Regn split & merge.

Detect^n of discontinuities

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$R = w_1 z_1 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

Point detect^n: $R = \sum_{i=1}^{9} w_i z_i$

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$ → Laplacian mask

$|R| \geq T$ → pt detected @ center

↳ T is a non-ve threshold

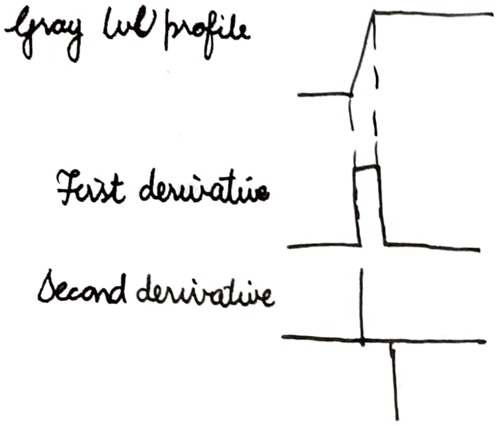
Line detect^n

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

Horizontal line detect^n	+45° line detect^n	Vertical line detect^n	-45° line detect^n
R_1	R_2	R_3	R_4

Apply all 4 masks on the image,
If at a certain pt $|R_i| > |R_j| \ \forall i, j$ then that pt is said to be associated with ~~line~~ mask i (more likely)

The pts left are the strongest responses, which, for lines one pixel thick, correspond ~~to~~ closest to the direct^n defined by the mask.



-ve 2nd der → bright reg^n
+ve " " → dark reg^n

TB Fig 10.7 & before.

→ A little amt of noise can have significant impact on the two key derivatives used for edge detect^n in images.
→ Image smoothing should be don before using derivatives for an image where noise is likely to be present.

Thresholding

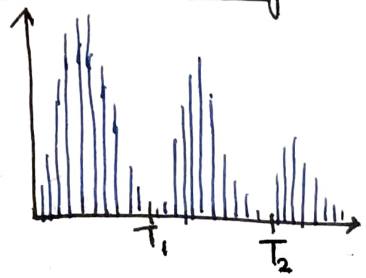
Purpose → divide an image into non-overlapping regions

A dark bckgnd & light object.

Segmentatn based on thresholding:—

> Intensity values are different in different regions.
> With each region, which represents the object, the intensity values are similar.

Global thresholding



$T_1 < f(x,y) < T_2$ → object 1

$T_2 \gtrless f(x,y)$ → object 2

$f(x,y) < T_1$ → background.

$f(x,y) = i(x,y)\, r(x,y)$

illuminatn fn, reflectance fn

Slide 13 → Example.

→ Select an estimate for T initially.

→ Segment the img using T, G_1 → graylevel > T ; G_2 → graylevel < T

→ Find avg gray level values ie m_1 & m_2 in G_1 & G_2.

→ Find $T_{new} = \frac{1}{2}(m_1 + m_2)$

→ Repeat the process till $T_i - T_{i-1} < \epsilon$; where ϵ is a small number.

Otsu's method (1979)

> Segmentatn based on "region homogeneity". Region homogeneity can be measured using variance
> Otsu's thresholding chooses the threshold to min. the intraclass variance of thresholded black & white pixels.

→ Find the th. that minimizes the weighted within class variance.

→ This turns out to be the same as maximizing the b/w class variance.

→ Operates directly on gray level histogram so it is fast.

$$P_1(T) = \sum_{i=0}^{T} p(i) \quad \& \quad P_2(T) = \sum_{i=T+1}^{L-1} p(i)$$

$p(i)$ → normalized freq. of i^{th} intensity

$$P_1(T) + P_2(T) = 1$$

B/w class variance: $\sigma_B(t) = P_1(t)\,[m_1(t) - m_G]^2 + P_2(t)\,[m_2(t) - m_G]^2$

Indiv. class variance: $\sigma_1^2(t) = \sum_{i=1}^{t} [i - m_1(t)]^2 \frac{P(i)}{P_1(t)}$; $\sigma_2^2(t) = \sum_{i=t+1}^{L-1} [i - m_2(t)]^2 \frac{P(i)}{P_2(t)}$

where $m_1(t) = \sum_{i=0}^{t} \frac{i\,P(i)}{P_1(t)}$; $m_2(t) = \sum_{i=t+1}^{L-1} \frac{i\,P(i)}{P_2(t)}$ → $P_1 m_1 + P_2 m_2 = m_G$ ⇒ $\boxed{m_G = \sum_{i=0}^{L-1} i\,p(i)}$

Pick the value that minimizes $\sigma_B^2(t)$ throughout the image.

For faster calc. : $\boxed{\sigma_B^2(t) = P_1(t)[m_1(t) - m_G]^2 + P_2(t)[m_2(t) - m_G]^2}$

Example : slide 24

<u>Drawbacks</u>:-

> Assumes histogram is bimodal (2 classes)

> Method breaks down when hist. ~~is ver~~ classes are very unequal (i.e., the classes have very different sizes)

→ $\sigma_B^2(t)$ may have 2 maxima

→ correct maxima may not be global max

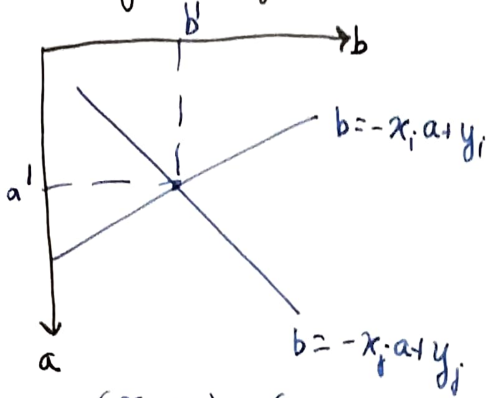
> Doesn't work well with variable illuminatn

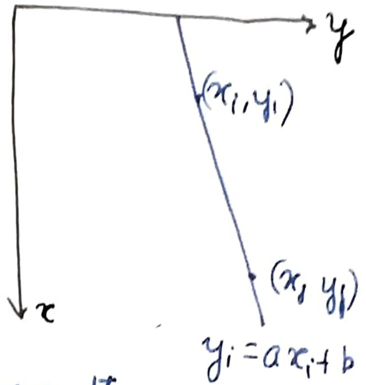
K	0	1	2	3	4	5	6	7
p_i	0	0	$1/16$	0	$1/16$	$6/16$	$6/16$	$2/16$
$P_1(T)$; $P_2(T)$	0, 1	0, 1	$(\frac{1}{16}, \frac{15}{16})$	$(\frac{1}{16}, \frac{15}{16})$	$(\frac{2}{16}, \frac{14}{16})$	$(\frac{8}{16}, \frac{8}{16})$	$(\frac{14}{16}, \frac{2}{16})$	$(\frac{16}{16}, \frac{0}{16})$ 1, 0
$m_1(T)$; $m_2(T)$								
m_G								
Variance interclass σ^2	0	0	0.759	0.759	0.80	0.76	0.37	0

$$\begin{matrix} 2 & 7 & 6 & 6 \\ 5 & 6 & 5 & 5 \\ 6 & 5 & 5 & 6 \\ 7 & 6 & 4 & 5 \end{matrix}$$

26/10/20 - L29

Hough Transform (1962) → A method to detect lines (or other parametric objects)





(x_i, y_i) & (x_j, y_j) → are collinear pts

MATLAB f^n
hough()

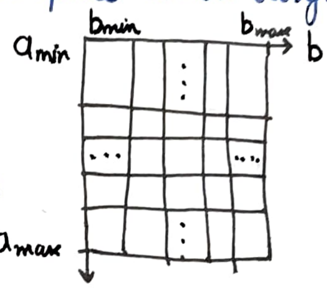
Procedure

→ Divide the parameter space into cells.

→ Plot parameter space lines corresponding to all pts (x_k, y_k)

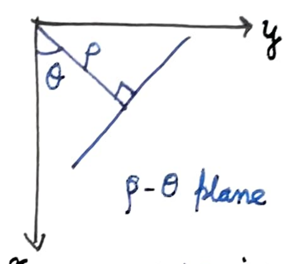
→ Identify the pts in ~~which~~ parameter space where large numbers of parameters space lines intersect.

Problem: vertical line → $a = \infty$



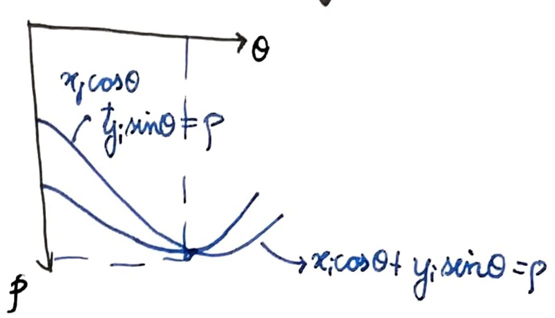
For ρ-θ plane

Pairs of θ & ρ will look like a sine wave. It's done for all a pt. locat^n



$x\cos\theta + y\sin\theta = \rho$

Fig 10.32 TB



Procedure

> For each data pt., a number of lines at diff ∠ are plotted through it @ diff angles.

> For each solid line a line is plotted which is ⊥^r to it & which intersects the origin

> The length & and angle of each dashed line is measured.

> Repeat for each data pt.

~~Count~~ ρ & θ at intersect^n is noted & the count of the corresponding bin is inc by 1.

In the end if ~~there are~~ value of a bin is P, then it means P pts in x-y plane lie on the line $x\cos\theta_j + y\sin\theta_j = \rho_i$

Drawbacks:

Computationally very expensive

For n pts in an image $\rightarrow \frac{n(n-1)}{2}$ possible lines

Then we have to do $(n-2)\frac{n(n-1)}{2}$ computat'ns /comparisons to see each of those 2 pts lie on some line joining the other 2 pts.

Applicat'ns: Check slides & TB.

27/10/20 − L30

> Revision of Hough transform

Segmentat^n by watershed algorithm

Based on visualizing an image in 3-D. Two spatial coordinates vs gray levels

3 Types of pts:

> pts belonging to regional min.

> pts at which a drop of water would fall with certainty to a single min. (watershed or catchment basin)

> pts at which water would be equally likely to fall to more than one such min. (crest lines or watershed divide line)

> Watershed line form a connected path, thus giving cont. boundaries b/w regions

> Watershed segmentat^n is in the extract^n of nearly uniform (blob like) objects from the background.

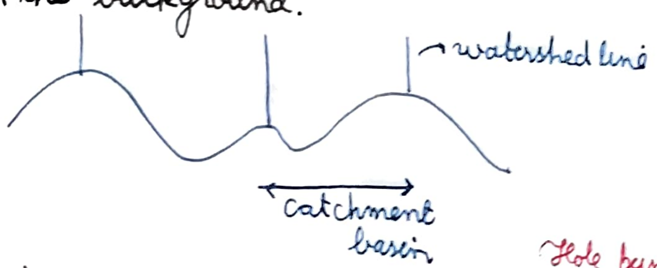


Image as a topological surface

All pixels in catchment basin are connected by a monotonically dec. path.

Procedure in slides along with an example.

Hole punched in @ minima & water floods in at a fixed rate

Marker based watershed → { image → threshold → dilat^n → watershed → edges

image → gradient magnitude → Processed

threshold or edges → markers → Processed

Processed ⟶ Final processed image.

Role of illuminat^n in segmentat^n

$f(x,y) = i(x,y) \cdot \underbrace{r(x,y)}_{}$ reflect

Fig 10.27 TB

Basic adaptive thresholding.

$$g(x,y) = \begin{cases} 1 & ; f(x,y) > T \\ 0 & ; f(x,y) \le T \end{cases}$$ → binary img.

$T = T\underbrace{[(x,y), p(x,y), f(x,y)]}$

Region based segmentat^n.

→ $\bigcup_{i=1}^{n} R_i = R$ (Every pixel in a rgn)

→ R_i is a connected rgn, $i = 1, 2, \ldots n$ (Pts in a rgn must be connected in some predefined sense)

→ $R_i \cap R_j = \phi$ for all i & j, $i \neq j$ (Rgns must be disjoint)

→ $P(R_i) = \text{TRUE}$ $\forall$ $i = 1, 2 \ldots n$. (The properties must be satisfied by the pixels in a segmented region)

→ $P(R_i \cup R_j) = \text{FALSE}$ $\forall$ $i \neq j$ (Rgns R_i & R_j are are diff in sense of predicate P)

$\boxed{29/10/20}$ - L31

Image processing

Acquisition → Enhancement → Restoratn → Segmentatn.

Image representatn & descriptn

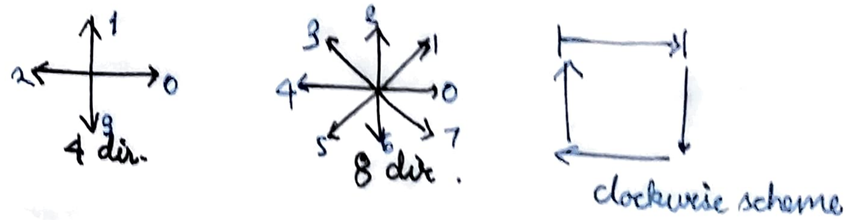
→ When features describe the shape, characteristic & it should be insensitive to variatn in size, translatn, rotatn.

→ when the features described enclosed region characteristics such as color, texture etc.

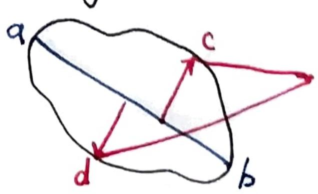
Shape representatn by using chain code

Chain code presents an object boundary by a sequence of straight line segments of specified length & direction

Examples in slides

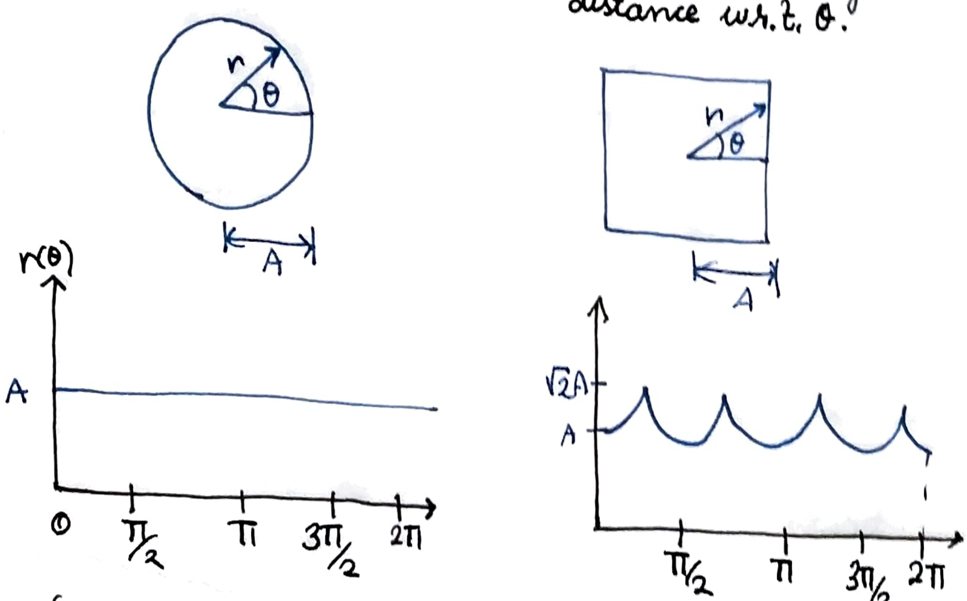


Splitting techniques



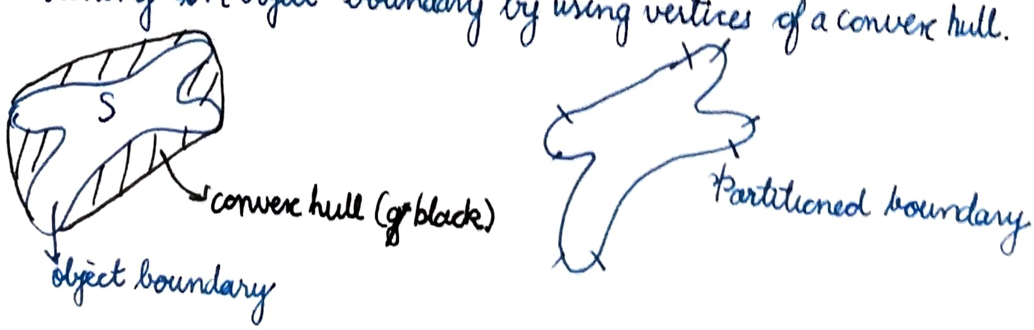
1. Find the line joining 2 extreme pts
2. Find the farthest pts from the line

Distance vs angle signatures : 2D object boundary in terms of a 1-D f^n of radial distance w.r.t. θ.



Boundary segments

Partitioning an object boundary by using vertices of a convex hull.



Objective → To represent & describe segmented or extracted feature info in an image in other forms that are more suitable than the image itself

Benefits → Easier to understand

Requires fewer units of memory, easy to understand & fast processing

Boundary descriptors

Simple geometric measures used to define a boundary.

→ simple descriptors → shape numbers → fourier desc. → statistical moments

Simple descriptors

> Length > Eccentricity > Diameters > Curvature

> Major-minor axis > Convex-concave > Basic rectangle > corner ($>90°$)

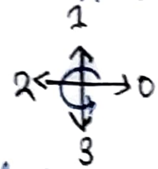
Length → no. of pixels along a boundary approx.

Size of circle or box (smallest) → that encloses the object

Curvature → rate of change of slope.

ACW diff → 1

CW diff → 3

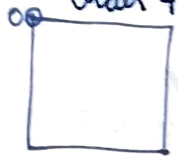


Shape numbers and order of feature →

Shape no → the first diff of smallest magnitude

Order 'n' → number of digits in seq.

Order 4

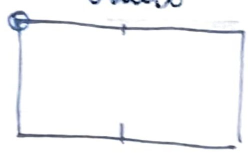


Chain code: 0 3 2 1

Diff: 3 3 3 3

Shape no.: 3 3 3 3

Order 6

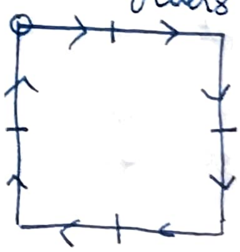


0 0 3 2 2 1

3 0 3 3 0 3

0 3 3 0 3 3

Order 8

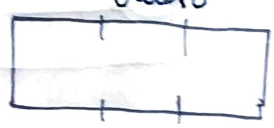


Chain 0 0 3 3 2 2 1 1

Diff 3 0 3 0 3 0 3 0

Shape 0 3 0 3 0 3 0 3

Order 8

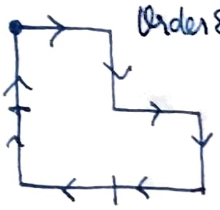


Chain 0 0 0 3 2 2 2 1

Diff 3 0 0 3 3 0 0 3

Shape 0 0 3 3 0 0 3 3

Order 8



Chain 0 3 0 3 2 2 1 1

Diff 3 3 1 3 3 0 3 0

Shape 0 3 0 3 3 1 3 3

Example 11.2

1 > Original boundary

2 > Find smallest rect that fits the shape.

3 > Create grid

4 > Find nearest grid.

Fourier descriptor

Coordinate as a complex no. (x = Real part, y = Imag. part)

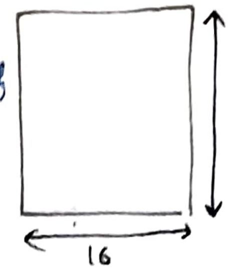
$$s(k) = x(k) + j\,y(k)$$

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-2\pi j uk/K} \quad \text{DFT}$$

Approx reconstruct^n from F. Desc. IDFT

$$\hat{s}(k) = \frac{1}{P}\sum_{u=0}^{P-1} a(u)e^{\frac{2\pi j uk}{K}}$$

Example 11.3 > $\underbrace{P = 2, 4, 8 \ldots 32 \ldots 64}_{\text{no. of samples Fourier coeff to reconstruct}}$, $K = 64$



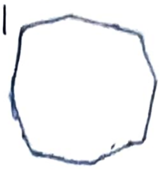
16

16

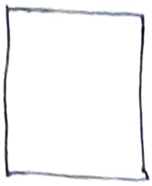
P = 2

P = 8

.... P = 61



P = 62



→ Approx the same.

Fourier Descriptor properties

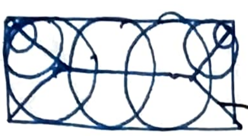
T/F	Boundary	Fourier descriptor
Identity	$s(k)$	$a(u)$
Rotat^n	$s_r(k) = s(k)e^{j\omega}$	$a_r(u) = a(u)e^{j\theta}$
Translat^n	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\,\delta(u)$
Scaling	$s_s(k) = \alpha\, s(k)$	$a_s(u) = \alpha\, a(u)$
Starting pt.	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

Skeletons

Reduce binary image objects to a set of thin strokes. Represent structural shape of a plane reg^n to reduce it to a graph. Retains important info abt the original object.

Medial axis transformation or SAT (Symmetry Axis Tf)

Medial axis is the locus of centers of maximal disks that fit within the shape.



loci

Prairie fire analogy → the boundary of an object is set on fire & the skeleton is the loci where the fire fronts meet & quench each other

Applicat^n : Skeletons

→ Skeletonizat^n is used in letter & character recog^n. Letters & chars are recog. by center lines of its strokes & is unrelated to width of the stroke lines.

Statistical moments

$r \rightarrow$ RV , $g(r_i) \rightarrow$ normalized RV.

The n^{th} moment : $\boxed{\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i)}$ where $\boxed{m = \sum_{i=0}^{K-1} r_i\, g(r_i)}$

First order moment : mean ; 2^{nd} order : Variance.

Order 0 : no. of pts in the data

Order 1 : ~~no~~ sum → used to get mean.

Order 2 : related to variance.

Order 3 : skewness of data.

→ Convert boundary segment into a 1D graph

→ View 1D graph as PDF f^n

→ Compute n^{th} order moment of the graph.

Region descriptors: Statistical properties

- Area of regn: no. of pixels in a regn
- Perimeter " " : Length of its boundary
- Compactness " " : (perimeter)^2 / area
- Mean & median of gray levels
- Min & max gray level values
- Number of pixels with values below & above the mean

Example: Slides (Infrared image of "अमेरिका")

→ For area.

$$A(R) = \sum_m \sum_n x(m,n)$$

→ number of pixels in the region.

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{array} \rightarrow \text{Area} = 5$$

→ For perimeter

$$P(R) = \sum_k \sqrt{[x(k)-x(k-1)]^2 + [y(k)-y(k-1)]^2}$$

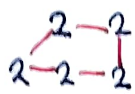
Perimeter = $1+1+1+1+\sqrt{2} = 4+\sqrt{2}$

→ Compactness → $C = \dfrac{P^2(R)}{A(R)}$

$$C = \frac{(4+\sqrt{2})^2}{5} = \frac{16+2+8\sqrt{2}}{5} = \frac{18}{5} + \frac{8}{5}\sqrt{2}$$

8×8 image in slides : same exercise

Feature '2' →

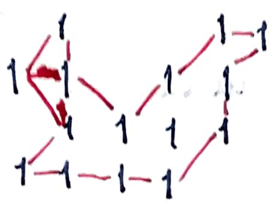


→ Area = 5

Perimeter = $4+\sqrt{2}$

$C(2) = \dfrac{18+8\sqrt{2}}{5}$

Feature '1' →



→ Area = 15

Perimeter = $3(1) + 3(1) + 8\sqrt{2}$
$= 6 + 8\sqrt{2}$

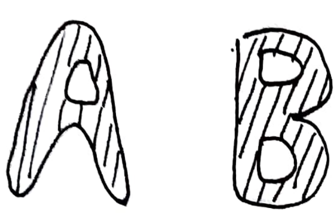
$C(\mathbb{1}) = \dfrac{36+128+96\sqrt{2}}{225}$

Topological descriptors:

- Property 1: the number of holes.
- " 2: the number of connected components
- " 3: Euler number : $E = C - H$ (#Connected component - # holes)

→ No. of holes = 2

→ No. of connected components = 3.



$E = 1 - 1$
$= 0$
$(C=1, H=1)$

$E = 1 - 2$
$= -1$
$(C=1, H=2)$

Property 4: Largest connected component (Example in slide)

Texture Descriptors

Texture → Smoothness or roughness of a surface
Types: Random, Regular

Random: described statistically, not by words or eqns. → Analyzed by stat. methods
Regular: " by words or eqns or repeating patterns with primitives (patterns)
→ Analyzed by structural or spectral (Fourier) methods.

$$\mu_n(z) = \sum_{i=0}^{K-1} (z_i - m)^n p(z_i)$$

z → Intensity
$p(z)$ → PDF or hist. of z.

$$m = \sum_{i=0}^{K-1} z_i \, p(z_i) \qquad \sigma^2(z) = \sum_{i=0}^{K-1} z_i^2 \, p_i(z_i)$$

> Relative smoothness: → $R = 1 - \frac{1}{1+\sigma^2(z)}$

> Uniformity → $U = \sum_{i=0}^{L-1} p^2(z_i)$

> Average entropy
$e = -\sum_{i=0}^{L-1} p_i(z_i) \log_2(p_i(z_i))$

Statistical approaches for Texture descriptors
Example: slide.

Regular ~~str~~ texture : Structural approach.

$S \rightarrow aS$ → here a represents the primitive element

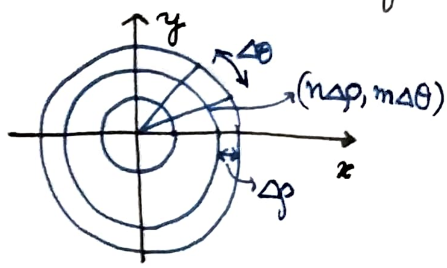
Rule for the structural approach

→ a a a a a a → string

Distance vs angle signatures

Yuan & Jang → Extractⁿ of rotatⁿ invariant signature based on practical geometry (IEEE 2001)

CPT → Central projectⁿ transform



$$f(\theta) = \sum_n f(\rho\cos\theta_m, \rho\sin\theta_m)$$

ρ → segmented in 'n'

θ → segmented into 'm' components

$\theta_m \in [0, 2\pi]$, $m = 1, 2, \ldots 360$

CPT done to extract contour; which can be then unfolded to get Y vs X variatⁿ

Principal components for descriptors

→ 1000 images of a subject, each one differs from each other ~~at~~ in some aspect. Then extracting principal components saves a lot of time & computatⁿ.

→ It reduces image data to smaller dim to represent image variability

→ Involves a procedure that t/f a number of possibly correlated variables into small number of uncorrelated variables called principal components

→ PCA: orthogonal linear t/f that transforms data to new coordinate system s.t. the greatest variance by any projectⁿ of data that comes to lie in first coordinate (1st PC), second greatest variance & so on.

Let $x = [x_1 \; x_2 \; \ldots \; x_n]^T$; mean: $m_x = E\{x\} = \frac{1}{K}\sum_{k=1}^{K} x_k$

Covariance matrix ~~m_x~~ $C_x = E\{(x-m_x)(x-m_x)^T\} = \frac{1}{K}\sum_{k=1}^{K} x_k x_k^T - m_x m_x^T$

Let $y = A(x - m_x)$; then $m_y = E\{y\} = 0$ & $C_y = A C_x A^T$

Find A s.t. → $C_y = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & & 0 \\ \vdots & & \vdots & & 0 \\ 0 & & \cdots & & \lambda_1 \end{bmatrix}$

⇒ $y = A(x - m_x)$ are uncorrelated. The component of y with greatest λ is called principal component.

Consider $x_1 = (0\,0\,0)^T$, $x_2 = (1\,0\,0)^T$, $x_3 = (1\,1\,0)^T$, $x_4 = (1\,0\,1)^T$

$$X = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow m_x = \frac{1}{4}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

$$\cancel{C_x = \frac{1}{4}\left(\begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} - \frac{1}{16}\begin{bmatrix} 9 & 3 & 3 \\ 3 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix}\right)}$$

$$\begin{bmatrix} cov(x_1,x_1) & cov(x_1,x_2) & cov(x_1,x_3) \\ cov(x_2,x_1) & cov(x_2,x_2) & cov(x_2,x_3) \\ cov(x_3,x_1) & cov(x_3,x_2) & cov(x_3,x_3) \end{bmatrix}$$

Cov > 0 → both dim inc together
Cov < 0 → one inc, other dec.
Cov = 0 → indep of each other

$$x - m_x \quad C_x = \begin{bmatrix} 0-3/4 & 1-3/4 & 1-3/4 & 1-3/4 \\ 0-1/4 & 0-1/4 & 1-1/4 & 0-1/4 \\ 0-1/4 & 0-1/4 & 0-1/4 & 1-1/4 \end{bmatrix} \begin{bmatrix} -\frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \Rightarrow \begin{bmatrix} 12/16 & 4/16 & 4/16 \\ 4/16 & 12/16 & -4/16 \\ 4/16 & -4/16 & 12/16 \end{bmatrix} = C_x$$

Object & Pattern Recognition (Chapter 12)

Norbert Wiener → " "

A pattern → arrangement of descriptors.

Pattern class → A family of patterns with common properties

Pattern recog^n → assign patterns to their respective classes.

→ Image indexing is done to reduce computat^{n}s in authenticat^n & verificat^n process.

→ Pattern template regularized to produce a few feature vectors s.t. embedding space of same subjects are close, while those belonging to different subjects lie far.

→ Clustering applied to get reps of index entries for index table.

→ Probe image presented to system, its feat vector is computed.

Patterns & features

Pattern: A set of consistent, characteristic form, style of an object; signature, color, shape, entropy.

Recog^n of patterns:

> Observe environment

> Learn to distinguish patterns of interest

> Make sound decisions & reasonable assignments of patterns to possible classes.

Three common pattern recog^n.

→ Vectors

Iris flowers

Edgar Anderson: Annals of Missouri Botanical garden, Vol 15, No. 3.

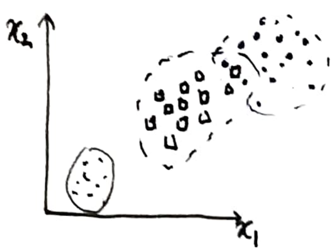
$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

x_1 → Petal width

x_2 → Petal length

x_3 → Sepal width

x_4 → Sepal length.



→ Strings

⇓

→ Vector patterns (Fig 12.2)

string descriptors: generate patterns of objects & entities whose structure is based on relatively simple connectivity of primitives, usually associated with boundary shape.

(Fig 12.3)

Fractal diagram

→ Trees

More powerful than string patterns. Hierarchical ordering schemes usually lead to tree structures.

(Fig 12.4, 12.5)

PCA → eigenphase.

Recognition based on decision

Decision theoretic approaches to recog^n are based on the use of decision f^{n}s.

$x = (x_1, x_2 \dots x_n)^T$ → n dimensional pattern vector. For W pattern classes, $\omega_1, \omega_2 \dots \omega_W$ we wanna find W decision f^{n}s $d_1(x), d_2(x) \dots d_W(x)$ with the property that:

If a pattern belongs to class ω_i, then: $\boxed{d_i(x) > d_j(x) \quad j = 1, 2, \ldots W; j \neq i}$

The decision boundary separating ω_i & ω_j is: $\boxed{d_i(x) = d_j(x) \text{ or } d_i(x) - d_j(x) = 0}$

(Fig 12.1)

Min distance classifier: $m_j = \frac{1}{n_j} \sum_{x \in \omega_j} x \quad j = 1, 2, \ldots N_c W$

$D_j(x) = \|x - m_j\| \quad j = 1, 2 \ldots W \quad ; \quad d_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j \quad j = 1, 2 \ldots W$

$$\boxed{d_{ij} = d_i(x) - d_j(x) = x^T(m_i - m_j) - \frac{1}{2}(m_i - m_j)^T(m_i + m_j) = 0}$$

↳ Surface given by this eqn is ⊥ bisector of line segment joining m_i, m_j.

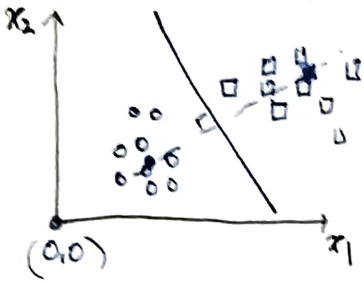
If $n = 2 \rightarrow$ line ; $n = 3 \rightarrow$ plane ; $n = 4$ or more $\rightarrow$ hyperplane.

Example:

$$d_1(x) = x^T m_1 - \frac{1}{2} m_1^T m_1 = 4.31 x_1 + 1.3 x_2 - 10.1$$

$$d_2(x) = x^T m_2 - \frac{1}{2} m_2^T m_2 = 1.5 x_1 + 0.3 x_2 - 1.17$$

where $m_1 = (4.3, 1.3)^T$ & $m_2 = (1.5, 0.3)^T$ $\longrightarrow$ $d_{12}(x) = d_1(x) - d_2(x) = 2.8 x_1 + 1.0 x_2 - 8.9 = 0$



Now say we have a pt $(x_1, x_2) = (3, \frac{1}{2})$

Then puttin' it in d_{12} we get

$2.8 \times 3 + 1.0 \times \frac{1}{2} - 8.9 \rightarrow < 0 \rightarrow$ Towards origin ✓

Theoretic methods: matching by correlatn

$C(x, y) = \sum_s \sum_t f(s, t) \, w(x + s, y + t) \quad x = 0, 1, 2 \ldots M - 1, \; y = 0, 1, 2 \ldots N - 1$

Refer Fig (12.8 TB)

$$\boxed{\gamma(x, y) = \frac{\sum_s \sum_t [w(s,t) - \bar{w}] \sum_s \sum_t [f(x+s, y+t) - \bar{f}(x+s, y+t)]}{\left\{\sum_s \sum_t [w(s,t) - \bar{w}]^2 \sum_s \sum_t [f(x+s, y+t) - \bar{f}(x+s, y+t)]\right\}}}$$

Correlatn coeff

Fig 12.9

Medical Imaging

Diff for each wavelength of light used for imaging.

[ ] → sampler → [ ] → quantizer → [ ]

Ultrasound
MRI
CT
PET

XRay
MRI
Nuclear medicine } em waves outside visible spectrum

Ultrasound → sound waves.

Interactⁿ of energy waves with tissues

> Absorption
> Attenuation
> Scattering.

$\boxed{23/11/20}$ - L37

Medical Image processing

X Ray Imaginary system

$1 PHz \to 10^{15} Hz$ 30 to 30,000 PHz. ($\lambda \sim 0.01$ to 10nm)

X Ray productⁿ using e⁻ collision (99% heat, 1% X rays)

Properties - in slides.

$I_d = I_0 e^{-\int \mu(x,y,z) dl}$ Attenuatⁿ coeff → $\mu(x, y, z) =$

24/11/20 - L38

Lecture slides