Image Processing

Visual Pattern Recognition (Sensory Organ - Eyes)

↓

Digital Image Recognition Systems → 3 Scenarios

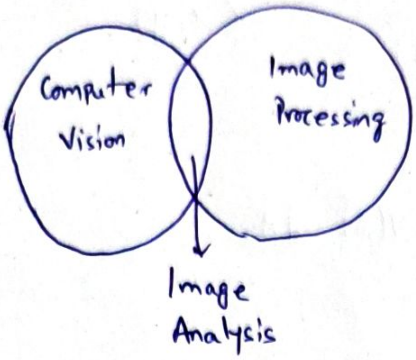
① INPUT : Text, Output : Image } Computer Graphics

② Input : Image, Output : Image } Image Processing

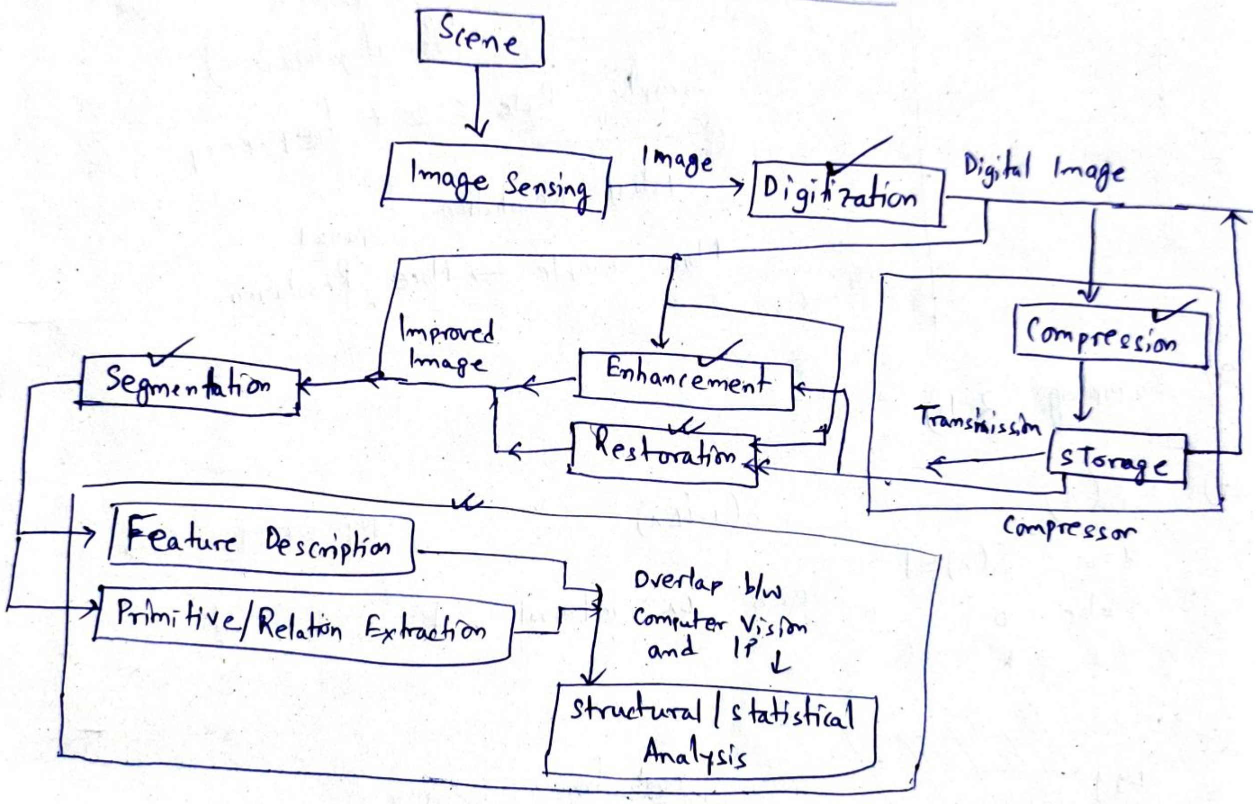
③ Input : Image, Output : Text / Image Description

↓

Image Analysis & Recognition OR Computer Vision



Overall Flow of Data in IP System.



Digital Image Formation

Real World Co-ord $P(u, v, w)$

a) light source - light falls on the scene $E(u, v, w, \lambda)$ falling on P (light falls of wavelength λ)

b) Reflectivity of P $R(\underbrace{u, v, w}_{\text{Material}}, \lambda)$

c) $\underbrace{C(u, v, w, \lambda)}_{\text{Camera Received}} = E(\quad) * R(\quad)$

d) Camera acts on $C(u, v, w, \lambda)$

① Projection 3d Real world coordinate $\longrightarrow$ Camera 3d Coordinate $\downarrow$ film co-ordinate (x, y)

② Camera Sensitivity $\longrightarrow V(\lambda)$

$$\text{to be stored} = \int_{\lambda-\Delta\lambda}^{\lambda+\Delta\lambda} C(u, v, w, \lambda)\, V(\lambda)\, d\lambda$$

$\Downarrow$ Continuous to digitization

Euclidean coordinate $\longrightarrow$ Integer Coordinate System (for digitization)

$\underline{\text{Sampling Rate}} \geq 2 * \text{frequency}$.

Spatial Resolution

More sample $\rightarrow$ More Spatial Resolution

e) Sampling (2-D)

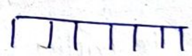
(1D) $\delta(x)$

$x = 0 \quad \delta(x) = 1$

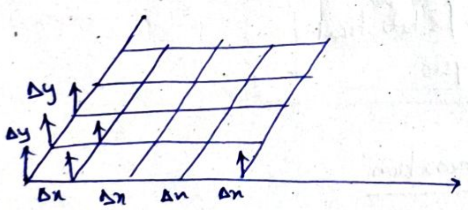
else $\quad 0$

$\delta(x - i\Delta x)$

each Δx interval, 1



Comb



comb fxn

$\downarrow$

$$\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \delta(x - i\Delta x,\; y - j\Delta y)$$

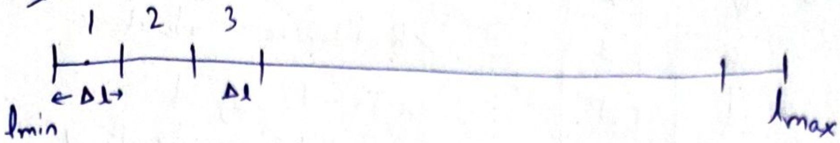
Real 2D film coordinate

$\hookrightarrow$ Integer Coordinate 2D (Pixel Coordinate)

$\Delta x, \Delta y \downarrow$, (Spatial Resolution) $\uparrow$, storage $\uparrow$

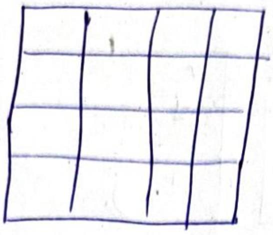
→ Quantization

$l_{min} + \frac{\Delta l}{2}$ (max Q. Error $\frac{\Delta l}{2}$)



k levels $\quad \dfrac{l_{max} - l_{min}}{k-1}$

2D Array $F(x,y)$

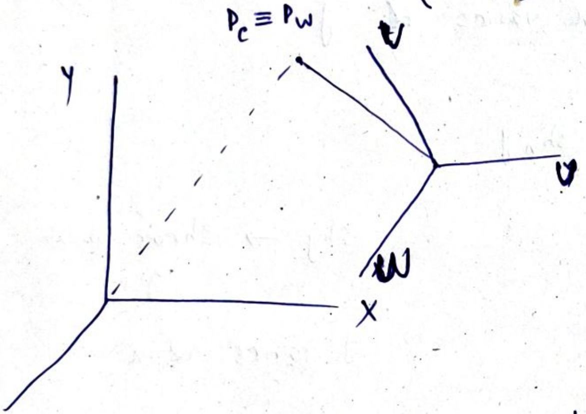


Projection

3D Real World (U,V,W) ⟶ 3D Camera Coord (X,Y,Z) ⟶ 2D Film Coord (x,y) → Pixel coord (i,j)

Perspective Projection → closer to camera, bigger

Sampling



→ Translate to map the origin

→ Rotation $(P_w - C)$

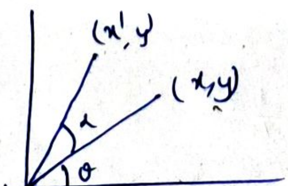
$(x,y) \xrightarrow{T_x, T_y} (x', y')$ Homogenous Coordinate

$x' = x + t_x$
$y' = y + t_y$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation



$x' = R\cos(\theta + \alpha)$

$y' = R\sin(\theta + \alpha)$

$x' = x\cos\alpha - y\sin\alpha \qquad y' = x\sin\alpha + y\cos\alpha$

~~Translation~~

Scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$S_x = 1 \quad S_y = -1$
↳ Reflection about X

$S_x = -1 \quad S_y = 1$
↳ Reflection about Y

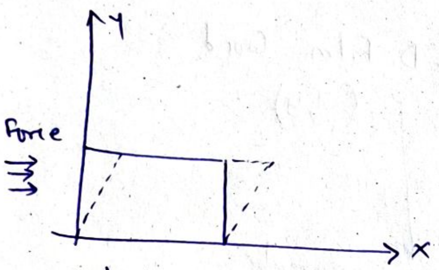
by θ

Rotation about origin

Combine Scaling and Rotation →

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x \cos\theta & -S_y \sin\theta & 0 \\ S_x \sin\theta & S_y \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear



for each point (x, y)
y co-ordinate same,
x varies $\propto$ y

sh_x

$x = x + sh_x y$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

sh_y → shear about y.

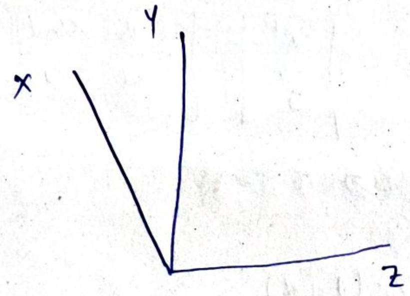
y varies $\propto$ x

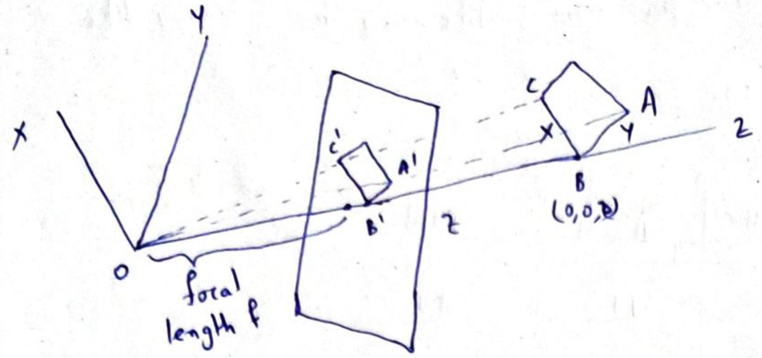
$y' = y + sh_y x$

3D → 2D
(Camera) (Film)

perspective projection

optic axis (Z - axis)





$B'C' = x$

$A'B' = y$

ΔOAB
$\Delta OA'B'$ $\quad \frac{y}{Y} = \frac{f}{Z} \Rightarrow y = \frac{fY}{Z}$

ΔOBC
$\Delta OB'C'$ $\quad \frac{x}{X} = \frac{f}{Z}$

$\Rightarrow x = \frac{fX}{Z}$

Raster graphics
↓
Pixel level
vector graphics
↓
following equation

$(X\ Y\ Z)$
↓
(x', y')

2D → homogenous Co-ord.
$(x, y, 1) \rightarrow kx\ ky\ k$

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

↳ Sampling → Quantization ⇒ $\underline{\text{2D Array}}$ → containing quantized intensity value.
M x N

Color image → for every pixel R G B will have different sensitivity value

<u>Image file</u>

<u>Image type</u>: Binary (bitmap) ↳ two level (B/W)
(BIN, ASCII)

BIN ← Gray Scale (graymap) → Intermediate levels
ASCII ↙ 8 bit : 256 values
0 ———— 255
B W

BIN ↖ RGB Colour Image (pixelmap) 8 bit Red Sensitivity
ASCII ↙ for each pixel 8 bit Green "
8 bit Blue "

(Colour exists but we can't detect) → pshycho visual Redundancy
↓
Removed while Compression

linux

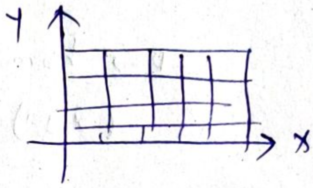
- .pbm → portable bit map
- .pgm
- .ppm

} file may be binary / ASCII

Image Type \ File type	ASCII	BIN
Bin	P1	P4
Gray	P2	P5
Color	P3	P6

file Header → Metadata

PX ← 1 digit no.



No. of col → x
No. of Row → Y

P1/2/...

optional { # ____
____

No. of cols (C) & No of Rows (R)

Max Poss. level Index ⇒

Header

Intensity for R×C pixels

If file stored in unreadable →

$ convert source dest.

extension

(color to grayscale
.3R + .6G + .1B
OR
$\left(\frac{R+G+B}{3}\right)$)

data[256][256]

```
for (i=0; i<256; i++)
for (j=0; j<256; j++)
    data[i][j] = j;
```

to modify such that both dimensions get doubled

$10 \times 10 \rightarrow 100$ points

$\Downarrow$

we need to make $20 \times 20 \rightarrow 400$ points

so $(x, y) \longrightarrow (x/2, y/2)$

Modified [x][y] = data[x/2][y/2]; instead 2*

Neighbourhood

4 - Neighbourhood
8 - Neighbourhood

x	✓	x	
✓	P(i,j)	✓	
x	✓	x	

P(i, j)

4 - Neigh
$(i-1, j)$
$(i+1, j)$
$(i, j-1)$
$(i, j+1)$

P & Q are two pixels

$d(P, Q) \Rightarrow$ distance / metric

if followings are satisfied

i) $d(P, Q) \geq 0$ if P & Q are same then it is 0.

ii) $d(P, Q) = d(Q, P)$

iii) $d(P, Q) \leq d(P, R) + d(Q, R)$ R is another pixel

Euclidean Dist

$P(x_1, y_1)$ $d_e(P, Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ (for Circle of Radius R)

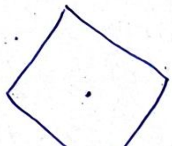
$Q(x_2, y_2)$

City-block | Manhattan Distance

$d_4 = |x_1 - x_2| + |y_1 - y_2|$

4 - Neighbours at distance $d_4 = 1$.

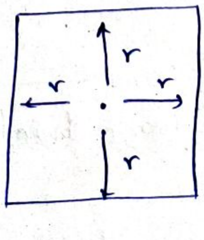
$\leq r$



diamond (diagonal = 2r+1)

Checkerboard Distance

$d_8 = \max\{|x_1 - x_2|, |y_1 - y_2|\}$



$(\leq r)$

$(2r+1) \times (2r+1)$

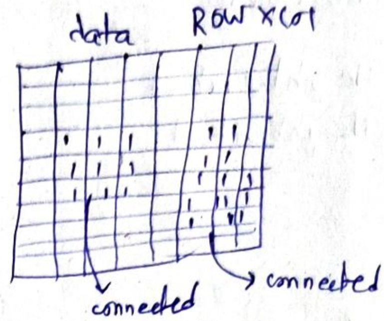
square

Path → There is a path b/w two pixels P & Q, if ∃ a sequence of pixels $P_1, P_2, P_3, \ldots, P_n$ such that $P_1 = P$ and $P_n = Q$, Intensity value of P_i is same as P & Q and $dist(P_i, P_{i+1}) = 1$

all 0 or all 1. for binary Image

distance can be anything.

\# Given a set of pixels, if for each pair of pixels, ∃ a path then the set is connected.



2 objects

generally consider
8-nbd → object
4-nbd → background

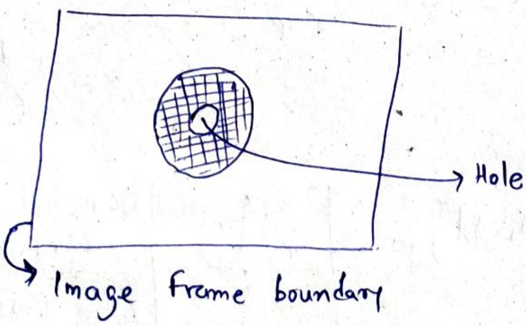
```
L=2
for(i=0; i<Row; i++)
for(j=0; j<COL; j++)
{
    if(data[i][j]==1)
        mark(data,i,j,L);
}
L++
}
```

(L-2) components.

```
void mark(int data[][61], int i, int j, int 2)
{   data[i][j]=L;
    if(data[i][j-1] ==1)
        mark(data, i, j-1, L);
    if(data[i][j+1]==1)
        mark(data, i, j+1, L);
    i-1, j.
    i+1, j
}
```

→ keep i, j limit

Identify
How many components and their size

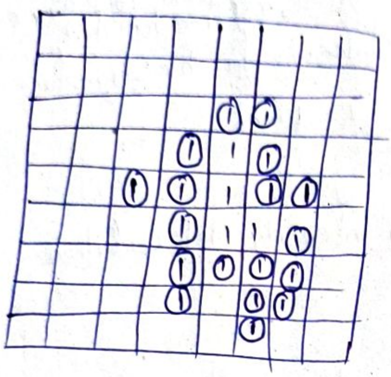


for a background pixel, if no path (wrt dist d_4) exists b/w the pixel and image frame boundary then its a hole element.

Maximally ~~ele~~ connected hole elements form the <u>hole</u>.

Boundary Pixels

↳ For an object A, the set of pixels for which atleast ~~pixel~~ one of its 8-neighbours (4-) not in A form the boundary.



$data[r][c] = 1$

(r, c) cnt = 0;

```
for (i = r-1 ; i <= r+1 ; i++)
for (j = c-1 ; j <= c+1 ; j++)
    if (data [i] [j])
        cnt++;
```

if cnt < 9 ↳ boundary pixel.

Linear Operation

$$H\big(a\underbrace{f_1(x,y)}_{\text{Images}} + b\underbrace{f_2(x,y)}\big)$$

↑ Operation,

linear if $H(af_1(x,y) + bf_2(x,y))$

$= H(af_1(x,y)) + H(bf_2(x,y))$ → additivity ~~operation~~ property → operation applied on sum of inputs = sum of ops ~~of~~ applied individually

$= aH(f_1(x,y)) + bH(f_2(x,y))$

Homogenuity property

↳ o/p of opn on constant multiplied by i/p = constant multiplied by o/p of opn applied on i/p.

Σ(Image) $= \Sigma\big(af_1(x,y) + bf_2(x,y)\big)$

↓ sum of pixels

$\equiv a\Sigma f_1(x,y) + b\Sigma f_2(x,y)$

$\max\{af_1(x,y) + bf_2(x,y)\}$ → Not linear if a, b diff sign

- Spatial domain operation
- Transformed Domain Operation

Spatial Domain Operation

Pixel operation
⇓
operation is applied on individual pixel

$T(F(R,C)) = 255 - F(R,C)$

Neighbour Operation hood
⇓
Pixel value is processed based on its neighbouring pixels

$f(x,y) = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} f(x+i, y+j)$

Geometric Transf / opn.
⇓
Rubber sheet transformation
① → Geometric Transformation of the pixel co-ordinate → (affine transf.)
→ Intensity for the transformed pixel
↓
② Intensity Interpolation

In the transformed image

$f'(x', y') \longrightarrow$ original $F(x,y)$

(may not be integer coord)

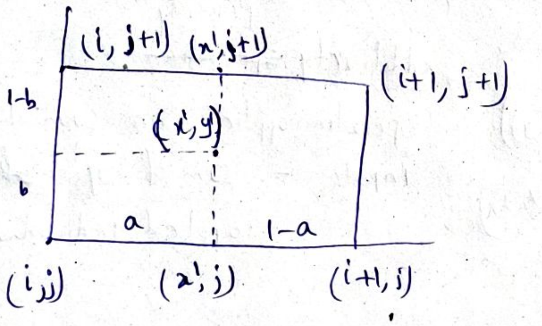
$F'(x', y') = F(\lfloor x + 0.5 \rfloor, \lfloor y + 0.5 \rfloor)$

→ ① Nearest Neighbour.

② Bilinear Interpolation
⇓

$(x', y') \Rightarrow (x, y)$

Original	Spatial
(x,y) ⟵	(x', y')
(may not be integer)	



③ Bicubic

$i \leq x' \leq i+1$

$j \leq y' \leq j+1$

assume,

$x' = i + a$

$y' = j + b$

Interpolating, in x dirn

$F(x', j) = (1-a) f(i,j) + a f(i+1, j)$

in Y dirn.

$F'(x', y') = (1-b) f'(x', j) + b f'(x', j+1)$

$f'(x', j+1) = af'(i+1, j+1) + (1-a) f'(i, j+1)$

$\Rightarrow$ 16 points $\rightarrow$ Bicubic Interpolation

① $\rightarrow$ Affine Tx for mapping
② $\rightarrow$ Interpolation for Intensity.

Transformed domain

spatial domain $f(x, y)$

$$\Big\downarrow \text{Transform}$$

$$\underbrace{T(u, v)}_{M \times N} = \underbrace{\sum\sum}_{M \times N} f(x, y) \underbrace{r(x, y, u, v)}_{\text{2d Transformation kernel}}$$

Reverse Transform

$$f(x, y) = \sum_u \sum_v T(u, v)\, s(x, y, u, v)$$

$\Rightarrow$ Separability

$\underbrace{r(x, y, u, v)}_{\text{2D Kernel}} \longrightarrow$ separable if

$r(x, y, u, v) = r_1(x, u) \cdot r_2(y, v)$ as if 1D

$\rightarrow$ Symmetric if

$r_1(x, u)$ and $r_2(y, v)$ are functionally same.

i.e $r_1(y, v) = r_2(y, v)$

First Fourier Transform

$$r(x, y, u, v) = e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

→ separable

→ symmetric

IFT

↳ $s(x,y,u,v) = e^{-j2\pi\left(\frac{ux}{M}\right)} \cdot e^{-j2\pi\left(\frac{vy}{N}\right)}$

↳ Reverse Tx.

Intensity Transformation

$s = T(r)$

o/p — Tx — i/p Intensity

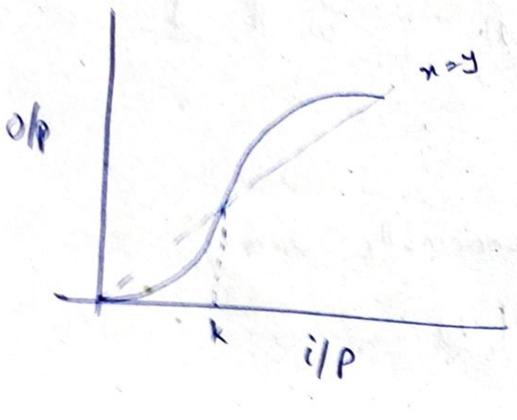
→ Pixel based

$T(r) = 255 - r$

If $T(f(x,y))$ is independent of x, y

↳ global transform

else → local transform



Noise

↓

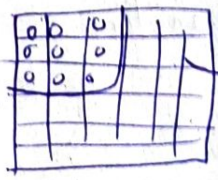
High freq.

(dirty part b/w clean easily noticable)

High freq filter

OR

FFT →



→ Relatively Higher freq

↓ IFT

get high freq

↳ Neighbour based

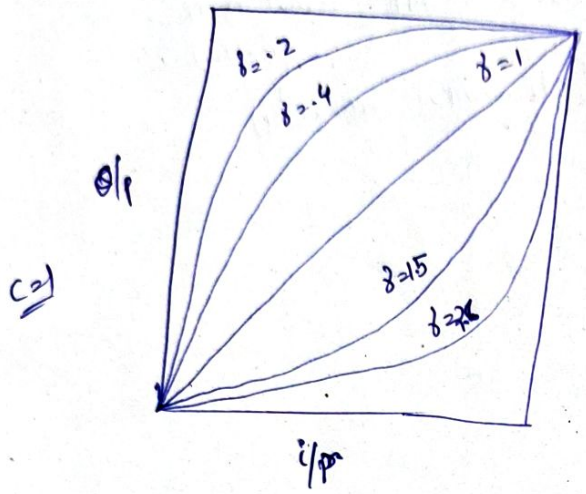
$s = c \log(1+r)$

s → o/p, c → constant, r → i/p

↳ brightness will be enhanced

exponential

↳ brighness ↓



Power law / Gamma Transform

Correction

↳ electroNic devices follow this.

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

s → o/p, c → constant, r → i/p

For Intensity Voltage transformation

Normally for CRT, $1.8 \leq \gamma \leq 2.5$

High Range of i/p → maps to low range of i/p (darker)

$\gamma < 1$ → logarithmic (nth root)

$\gamma > 1$ → Exponential (nth power)

→ If we want to correct, we will use $1/\gamma$

↳ γ correction.

Image Enhancement

Conversion of image from one form to another

→ Capturing
→ Digitization } → ↳ Degradation

point spread fun.

Transmit
Compress }

f (original undegraded)

$$g(x,y) = H\, \underbrace{f(x,y)}_{\text{original}} + \cancel{\eta}$$

ignore

↳ Noise

H → Point spread fun (focal length, material, etc)

but H → unknown

⇓ (Appln. specific)

Ad hoc mechanisms to satisfy the requirement of the application.
(certain aspects otherwise not visible → prominent)

Degradation model (known)

↓ we will apply inverse

⇒ Image Enhancement.

Image Enhancement

⇒ Ad hoc mechanism to reveal some data reqd. in the appln. which is otherwise not available

i/p Image $\xrightarrow{\text{processing}}$ Enhanced Image

⇑

Metric to judge
- → Subjective / Qualitative (User feedback)
- → Quantitative (SNR signal to Noise Ratio)

SSIM (Structural similarity)

Enhancement
- → Contrast Improvement
- → Noise cleaning
- → Image sharpening

Contrast Improvement

- Improper ~~[illegible]~~ Illumination
- Aperture is not proper
- shutter speed not proper

⇒ Contrast may be compromised

Intensity Histogram

Shape and range of histogram reflects a lot.

poor contrast → Roughly we can say (Due to noise)

(max I - min I) is less , poor contrast

↓

dynamic range of Intensities is confined in a small range.

In case of noise, we can say that Intensities < k % of total → noise

⇓

Enhance

↳ dynamic range should cover the available / allowable intensity range (Histogram stretching)

Histogram stretching

→ linear stretching

$m \rightarrow$ i/p Intensity

$T(m) = l$

↓

Transformation.

- $T(m)$ where $m \in [m_{min}, m_{max}]$

↳ $l = T(m)$ must be $\underbrace{[l_{min}, l_{max}]}_{\text{allowable}}$

- for any $m_2 > m_1$

monotonically increasing.

$T(m_2) \geq T(m_1)$ → cannot properly map reversely

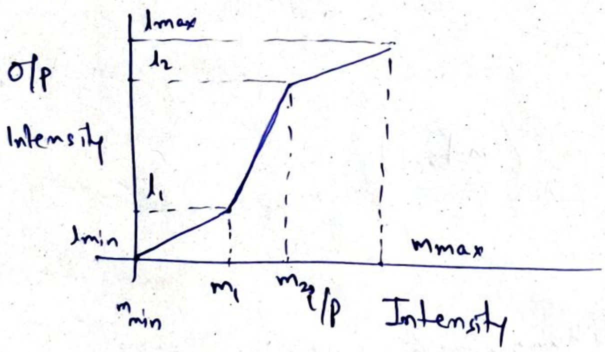
10 → 12

11 → 12

12 → 11 / 10 ?

$$T(m) = \left(\frac{l_{max} - l_{min}}{m_{max} - m_{min}}\right)(m - m_{min}) + l_{min}$$

→ Piecewise Linear Stretching.



we want to enhance m_1 to m_2 part

to increase range of coverage

If $(m \leq m_1)$ $\quad \dfrac{l_1 - l_{min}}{m_1 - m_{min}}(m - m_{min}) + l_{min}$

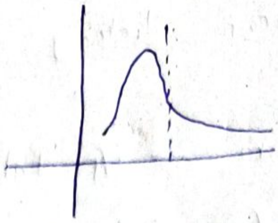
If $(m_1 < m \leq m_2)$ $\quad \left(\dfrac{l_2 - l_1}{m_2 - m_1}\right)(m - m_1) + l_1$

If $(m \geq m_2)$ $\quad \left(\dfrac{l_{max} - l_2}{m_{max} - m_2}\right)(m - m_2) + l_2$

Logarithmic Stretching

Majority Pixels are occupying a narrow lower Intensity range

→ Dark Image



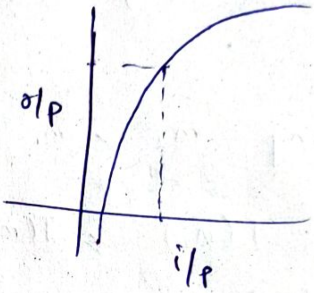
⇓ to enhance contrast

lower intensity values are more stretched at the cost of compressed stretching of higher Intensity value.

$$T(m) = (I_{max} - I_{min}) * \frac{\log(m - m_{min} + 1)}{\log(m_{max} - m_{min} + 1)} + I_{min}$$

If any captured device follows exponential characteristics

↳ apply logarithmic to get better.



Exponential stretching → opposite

Weber's Ratio →

A change of brightness δB at a point with its nbd average Intensity B

$\frac{\delta B}{B}$ is weber's Ratio. (constant)

(Surroundings dark → small change noticeable
— bright → big change need to notice)

→ To make change noticable,

δB ↑ with B ↑

i/p Image

$[m_{min}, m_{max}] \longrightarrow [l_{min}, l_{max}]$

No. of levels $= m_{max} - m_{min} + 1$

$\delta \rightarrow$ weber's Ratio

$$l_{i+1} = l_i + \delta l_i$$

webers ratio +1

$$\frac{l_{i+1}}{l_i} = \frac{l_2}{l_1} = \frac{l_3}{l_2} = \cdots \frac{l_k}{l_{k-1}} = c$$

$$l_k = c^{k-1} l_1$$

$$c = \left(\frac{l_k}{l_1}\right)^{1/k-1} = \left(\frac{l_{max}}{l_{min}}\right)^{\frac{1}{m_{max}-m_{min}}}$$

$$T(m) = l_{min} \cdot c^{(m-m_{in})}$$

$$= l_{min} \cdot \left(\frac{l_{max}}{l_{min}}\right)^{\frac{1}{m_{max}-m_{min}}}$$

$m_{min} \rightarrow l_{min}$

$m_{min}+1 \rightarrow c\, l_{min}$

$\vdots$

$m \rightarrow c^{m-m_{in}} \cdot l_{min}$

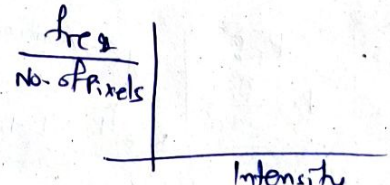
$\searrow$ following perception (weber's ratio)

Contrast Enhancement

$\rightarrow$ Histogram Stretching

Histogram Equalisation

$\rightarrow$ Intensity Histogram



$\rightarrow$ Normalize the histogram

Intensity values $r \Rightarrow$ random variable

pdf of $r = P_r(r)$

$s = T(r)$ — Transformation

s → Random variable

Say, $P_r(r) \rightarrow$ known

$T(r) \nearrow$

$P_s(s) \rightarrow ?$

There are L levels $\rightarrow L-1$ intervals

$$T(r) = (L-1)\int_0^r P_r(w)\, dw$$

Tx fxn of our interest.

$$T(r) = (L-1)\underbrace{\int_0^r P_r(w)\,dw}_{cdf}$$ w is the dummy variable of integration

- $\rightarrow$ Non-Negative
- $\rightarrow$ 0 to 1

$T(r) \rightarrow [0, L-1]$

$T(r_1) \leq T(r_2)$ if $r_1 < r_2$

$\hookrightarrow$ monotonically increasing Tx fxn.

$$P_s(s) = P_r(r)\left|\frac{dr}{ds}\right|$$

as if total area same $P_s\,ds = P_r\,dr$

$$T(r) = (L-1)\int_0^r P_r(w)\,dw$$

$$\frac{ds}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r P_r(w)\,dw\right]$$ (Leibnitz's Rule)

$$= (L-1) \cdot P_r(r)$$

differentiating definite Integral wrt upper limit $\equiv$ Integral evaluated at the limit

$$P_s(s) = P_r(r)\frac{1}{(L-1)P_r(r)}$$

$$\Rightarrow \boxed{P_s(s) = \frac{1}{L-1}}$$

Assuming $P_r(r)$ is continuous and differentiable

$\rightarrow$ All are equiprobable $\rightarrow$ uniform

$\Downarrow$

Equalisation

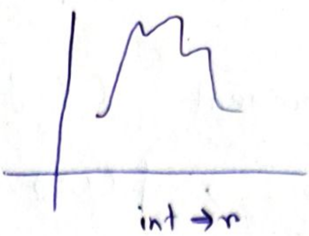
$m\times n \rightarrow$ image

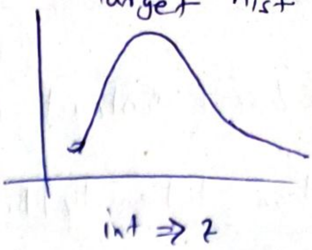
i/p Intensity	Count	Prob. $P_i = \frac{n_i}{mn}$	cdf $\rightarrow$ continuous $\sum_{k=0}^{i} P_k$	Transfer map $(L-1)c_i$ (Rounding)
0	n_0			
1	n_1			
2	n_2			
⋮	⋮			
$L-1$	n_{L-1}			

$r \rightarrow s$

$s = (L-1) \cdot$ cum. prob.

Histogram Matching / Specification





i/p Intensity $\quad P_r(r)$

$$s = T(r) = (L-1)\int_0^r P_r(w)\,dw$$

$r \rightarrow z$

another fn. on z

$$G(z) = (L-1)\int_0^z P_z(v)\,dv = s$$

$$z = G^{-1}(s)$$

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

T(r) monotonically increasing but not strictly so getting a inverse is not possible always

\# Perform histogram equalisation on i/p Image and form a mapping table

T_1

r	s

\# Target $P_z(z)$ given → Apply histogram Equalization

T_2

z	s

\# for a value of r, say r_k from T_1 get, get the corresponding s_k.

* Search in T_2 for s_k

→ If single match → get corresponding z

→ If multiple match

↳ select z as per some convention (say, smallest one)

Contrast Enhancement

↳ Global Approach (till now, we have done globally)

Local contrast Enhancement (x, y)

Nbd Mean Intensity / Brightness → $\bar{f}(x,y)$

global brightness → $\bar{f}$

Intensity variance over the nbd $\sigma^2(x,y)$

↳ Indicator about local contrast

→ Scaling factor

$$K\left[f(x,y) - \bar{f}(x,y)\right] + \bar{f}(x,y)$$

$$\underbrace{f'(x,y)}_{\text{corrected value}} = k\left(\underbrace{f(x,y) - \bar{f}(x,y)}_{\text{local contrast}}\right) + \bar{f}(x,y)$$

$$k = c * \frac{\bar{f}}{\sigma^2(x,y)}$$

$0 \leq c \leq 1$

to increase contrast

if already high → less

less → high

If $\sigma^2(x,y) \geq th$ → $f'(x,y) = k\left(f(x,y) - \bar{f}(x,y)\right) + \bar{f}(x,y)$

else $f'(x,y) = f(x,y)$

Filter / kernel

Spatial Filter

Applying a filter on the pixel

⇓

It's a sum of product operation

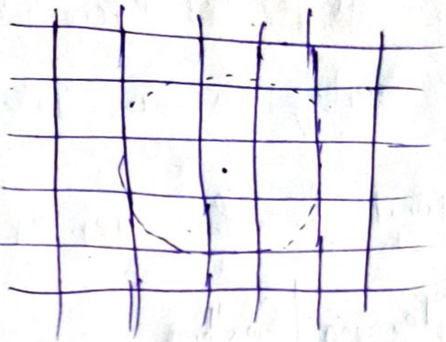
size of the filter / kernel ⇒ Defines the nbd

coeff of the kernel ⇒ Operations being done

Spatial kernel ⇒ 2D Matrix

w_i's are coeff

w_1	w_2	w_3
w_4	w_0	w_5
w_6	w_7	w_8



f filtering at (x,y)

w

$(2a+1)(2b+1)$

$$f'(x,y) = \sum_{i=-a}^{a}\sum_{j=-b}^{b} w_{ij} * f(x+i, y+j) = \sum_{0}^{2a}\sum_{0}^{2b} w_{ij} * f(x-a+i, y-b+j)$$

Carry out the operation at each point

$\Downarrow$

→ Image is convolved with a the kernel ⇒ This is convolution.

Kernel $m \times n$

2D-kernel is separable

$h(x,y) = h_1(x) \times h_2(y)$

if Rank = 1

$W = v \cdot w^T$ $\quad v, w$ $\quad v \to m\times 1$, $w \to n\times 1$ vectors

Inner Product of v, w

$m \times n$ matrix

$$W \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad v = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad w = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Rank = 1 → separable

mn → kernel $\qquad M \times N$ → Image Size

No. of operation → $mnMN$

if separable → $\underline{v}$ mMN $\quad \underline{w}$ nMN

$= MN(m+n)$

$(m+n) << mn$

Spatial Filter Designing / Defining

→ based on Mathematical formula
(Say Avg, Differentiate, Integration etc)

→ by sampling a 2D spatial fxn of special Interest.

→ Spatial fxn for freq rsp filtering

Noise Cleaning / Smoothing

Smoothing is done to get rid of spurious ~~noise~~ noise & false Contour.
Blurring

May Arise because of digitization process / transmission / Quantisation and their combinations.

Noise → Zero Mean, Pairwise Independent

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

Unknown

Degradation Model

Noise at a pixel not influenced by noise at another pixel

Image Averaging

$f(x,y)$ transmitted no of times

$g_i(x,y) = f_i(x,y) + \eta_i(x,y)$

$$\frac{1}{n}\sum_{i=1}^{n} g_i(x,y) = \frac{1}{n}\sum f_i(x,y) + \frac{1}{n}\sum \eta_i(x,y)$$

if n is large ⇒ 0 (zero mean)

→ Simple method

→ Problem → Correspondence b/w pixels (Image Registration) among $g_i(x,y)$ is required.

Mean filter / Box filter   (Any filter → Convolution)

$(2k+1)(2k+1)$ ⇒ all weights are say, 1 w_i

Rectangular (Box)

$$\hat{g}(x,y) = \left(\frac{1}{(2k+1)*(2k+1)}\right) \sum_{i=-k}^{k} \sum_{j=-k}^{k} 1 \cdot g(x+i, y+j)$$

(avg. Intensity of nbd)
avg → Mean

→ Noise characteristics → Symmetric, zero mean, pairwise uncorrelated

Normalization → Sum of Intensities in the nbd remains same

Based on the assumption

↳ Consistency of Intensities in the nbd

↳ Filtered value at a point is nothing but more pertinent value ~~keeping~~ w.r.to nbd.

3x3 kernel

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

3x3 kernel convolves with itself

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

→ 5x5 Kernel
Weighted Mean
Symmetric

If any symmetric kernel with non-negative weights | away from center, → weights decreases monotonically

↳ Can be used for smoothening.

↓

Weighted Mean Filter.

$w \; (2k+1) \times (2k+1)$

$$\hat{g}(x,y) = \frac{1}{\sum\sum w(i,j)} \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) * g(x+i, y+j)$$

Ordered Statistics filter

2x2 with 2x2

$$\begin{matrix}1&1\\1&1\end{matrix} \rightarrow \begin{matrix}1&2&1\\2&4&2\\1&2&1\end{matrix}$$

(3x3)

Mean / Weighted filler

↳ Filter / Coeff of kernel

depends on position of element in nbd

↳ Linear combination of Nbd intensities.

↳ Blurs the good edges also

→ preserves good edges to some extent

ordering of Filtered value depends on the position of intensities of nbd pixels.

↳ edge preserving to some extent.

ordered list of Intensities → In which position

Median filter

Intensities of nbd (kernel size) are sorted

$v_1\ v_2\ --\ v_n$ (ordered)

$$\hat{g} = \frac{1}{\sum w_i} \sum w_i v_i \qquad w_i = 1 \text{ if } i = n/2$$

else $w_i = 0$

$$\hat{g}(x,y) = \frac{1}{\sum\sum w(i,j)} \sum\sum w(i,j)\ g(x,y)$$

Impulsive noise (salt & pepper noise)

salt ↓ white

pepper ↓ black

↳ Sort → get Rid of both white / black

Sort → time Consuming.

Mid-Range filter

→ Soft and unaffected tails (extremes)

Mid range values are affected. → opposite of this

$w_i = 1$ if $1 \leq i \leq k$

OR $n-k < i \leq n$

else $w_i = 0$

$\Sigma w_i = 2k$

⇒ trimmed Mean Filter.

→ Fat tail

$w_i = 1$ if $k \leq i \leq n-k$

else 0

$\Sigma w_i = n-2k$

Max / Min filter

$\hat{g}(x,y) = \max\{x_i\}$

OR $\min\{x_i\}$

In case of Salt & Pepper Noise

Max followed by MIN | ⇒ Max MIN filter

Min followed by MAX | ⇒ Min Max filter.

(Max - Min) (Min - Max) → cascading number of times

⇓

Median filter.

Consecutive Max - Min → No effect

Consecutive Min - Max → No effect

low pass

↳ Smoothening.

Gaussian low pass filter (spatial)

Box filter ⇒ Blurring operation is more biased in perpendicular dirn.

Isometric → Independent of orientation

→ Gaussian kernel ⇒ Symmetric in circular Manner.

$$G(x,y) = k\, e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$k \downarrow$ constant

$$k e^{-\frac{r^2}{2\sigma^2}}$$

$r \rightarrow$ radius

Increases Radially outward

→ same Radius → same weight.

$\sigma \rightarrow$ scale of blur

beyond $\mu \pm 3\sigma$

→ coeff = 0

Freq Domain

$$f(x,y) \xrightarrow{FFT} F(u,v)$$

$f(x,y) \rightarrow M \times N$, $F(u,v) \rightarrow M \times N$

$$F(u,v) = \sum_x \sum_y e^{-2\pi j\left(\frac{ux}{M} + \frac{vy}{N}\right)} f(x,y)$$

Inverse Transform

$$f(x,y) = \frac{1}{MN} \sum_u \sum_v e^{2\pi j\left(\frac{ux}{M} + \frac{vy}{N}\right)} F(u,v)$$

→ Unless image is a very simple one, It is very difficult to relate a freq component to spatial domain Image.

→ Lowest freq Comp

$F(0,0) \Rightarrow$ Avg. Brightness

every freq domain component → A diff linear combination of all spatial domain components

As we move on $F(u,v)$ ↑es.

Freq corresponds to Rate of Intensity Change in spatial Domain.

0-10r → High F
0-10 → low F

$F(u_1, v_1)$

$F(u_2, v_2)$ → High Freq component

$u_1 + v_1 < u_2 + v_2$

$f(x,y) \xrightarrow{FFT} F(u,v)$

$H(u,v)$ ⇒ Freq Domain Filter

$F(u,v)\ H(u,v)$ ⇒ Element wise product

$\downarrow \qquad \downarrow$

$f(x,y)\ h(x,y)$ Spatial convolution

IFT

$f^{-1}(F(u,v) \times H(u,v))$ ⇒ final Response $\hat{f}(x,y)$

$h(x,y)$ → impulse response of $H(u,v)$ → corresponding spatial domain Tx $h(x,y)$

Impulse fxn → $\delta(x,y) \xrightarrow{FFT} 1$

particular Instance
↳ high else 0.

$\underbrace{FT(\delta(x,y))}_{1}\ H(u,v) = H(u,v)$

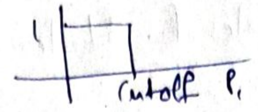
Known

Inverse

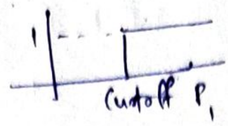
$h(x,y)$

Freq Domain Filters

Low Pass filter →

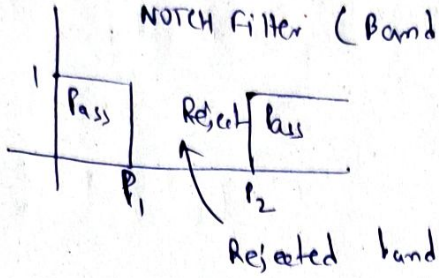


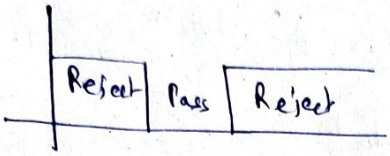
High Pass Filter →



Now,

NOTCH Filter (Band Reject Filter)





Band Pass Filter

Spatial

$lp(x,y)$ low pass filter

$\delta(x,y) - lp(x,y)$ High Pass filter

NF | BRF $\quad lp_1(x,y) + \delta(x,y) - lp_2(x,y) = BR(x,y)$

BPF $\quad \delta(x,y) - BR(x,y)$

1D Gaussian Filters

$H(u) = A e^{-\frac{u^2}{2\sigma^2}}$ $\quad \sigma \to$ Scale $\quad$ Spatial

$\Big\downarrow$ IFT

$h(x) = A e^{-2\pi^2 \sigma^2 x^2}$ $\quad$ ~~Time~~ freq

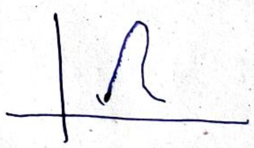
↳ Gaussian Pair (Both are gaussian)

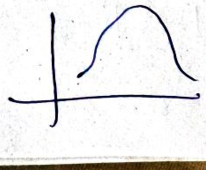
If the scale | spread is higher in tx domain,

response is narrower in freq domain.

↓

low pass filter





← time

wider → smooth out | blur larger areas

Ideal low Pass filter

$D(u, v)$
$\searrow$ Distance from center.

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

D_0 $\searrow$ Cut off freq

$$f(x,y) \xrightarrow{FT} F(u,v)$$

$M \times N$ $\quad$ $M \times N$

Gaussian Low Pass filter

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

$D \rightarrow D(u,v)$

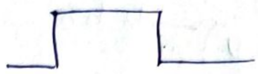
$D_0 \rightarrow$ cut off freq

Butterworth filter

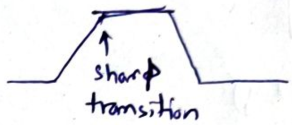
$$H(u,v) = \frac{1}{\left(1 + \left(\frac{D}{D_0}\right)^{2n}\right)}$$

$\downarrow$ Sharpness of low pass filter with less ringing.

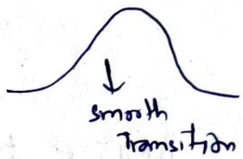
$n \uparrow \rightarrow$ LPF



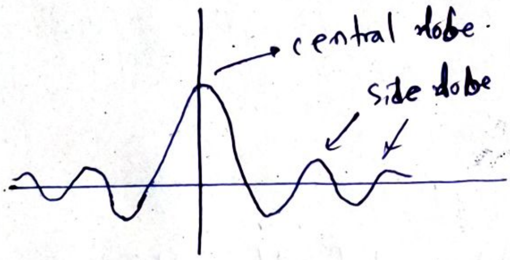
Mean filter



Gaussian filter



Spatial Domain filter (low Pass)



$\frac{\sin x}{x}$ (Sinc function)

Central lobe is responsible for blurring

Side lobes introduce artefacts $\longrightarrow$ Ringing

$\downarrow$

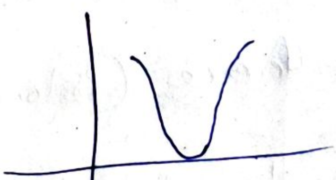
Undesired

Gaussian filter

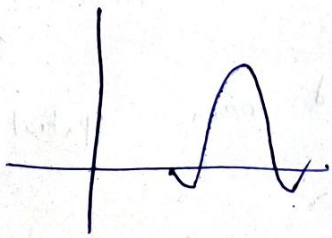
$\hookrightarrow$ Less blurring but no ringing.

$\downarrow$

High Pass filter $\rightarrow$



Spatial Domain



HPF

IHPF $= 1 - H_L(u,v)$

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

GHPF $= 1 - A e^{-\frac{D^2}{\Delta^2}}$

BW $\cdot$ $\dfrac{1}{1+\left(\frac{D_0}{D}\right)^{2n}}$

$H(u,v) \rightarrow$ Tx fn.

$$\delta(x,y) \xrightarrow{FT} F(\delta(x,y))$$

$$\Big\downarrow H(u,v)$$

$$\frac{F(\delta(x,y)) * H(u,v)}{1} \equiv H(u,v)$$

$H(u,v) \downarrow h(x,y)$

$$h(x,y) \xleftarrow{IFT} H(u,v)$$

Spatial Domain — Freq Domain

Bilateral Filtering

→ Noise Cleaning
→ Texture Editing

| preserves the edges → Relative diff b/w nbd Intensities

→ Filtered value at a pixel will be the linear combination of neighbourhood pixel Intensities.

→ wt. of ~~ab~~ neighbours not only depend on their Euclidean distance of neighbours but also their intensity differences (Photometric distance).

→ nbd means spatial proximity and photometric proximity

$$\hat{f}(x,y) = \frac{1}{W_p} \sum_{x_i} \sum_{y_i} I(x_i, y_i)$$

Sum of the wt. $(x_i, y_i) \in S_N$

$w \Rightarrow$ $f_s(\ )$ spatial kernel $\xrightarrow{\text{Scale } \sigma_s}$ $d_s = \sqrt{(x-x_i)^2 + (y-y_i)^2}$

$f_r(\ )$ Range kernel $\xrightarrow[\text{Scale } \sigma_r]{}$ $d_r = \cancel{\sqrt{}}\, |I(x_i, y_i) - I(x,y)|$

$f_s(x_i, y_i) * f_r(x_i, y_i)$

May be Gaussian / other

$f_s(x_i, y_i) = A_1 e^{-\frac{d_s^2}{\sigma_s^2}}$ → spatial gaussian (g_s)

Product of gaussian also gaussian.

$f_r(x_i, y_i) = A_2 e^{-\frac{d_r^2}{\sigma_r^2}}$ → Range Gaussian (g_r)

$f(x,y) - \hat{f}(x,y) = \text{Noise / Texture}$

If d_r ↑es → g_r is small

If d_s ↑ → g_s is small

g_r is high if intensity diff is small

is small if ―――― is high → edge preserving

g_s is low for spatially distant neighbours.

wt ⇒ is the product of two kernel coefficients

⇒ If one of them is low, the product is also low.

If σ_r ↑ ⇒ g_r approaches some constant.

↳ bilateral filter $\xrightarrow{\text{tends to}}$ spatial gaussian kernel

σ_s ↑ ⇒ ~~spatial scale~~ g_s → some constant

the amount of blurring

↳ bilateral filter → Range kernel

$\sigma_s \uparrow \rightarrow$ more blurring
if σ_r is low $\rightarrow$ Range kernel is narrow.

} overall effect is low.

$$f(x,y) \xrightarrow[\text{bilateral filter}]{} \hat{f}(x,y) \qquad f(x,y) - \hat{f}(x,y)$$

$\hookrightarrow$ Noise | Texture

Image Sharpening

Major degradation is in the form of blurring $\rightarrow$ good or bad pixels both

blurring $\rightarrow$ Sort of Integration

$\rightarrow$ differentiation

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

∇^2 $\leftarrow$ laplacian operator

$\downarrow$

Sharpened edges.

Taylor Series exp.

$$f(x+\Delta x) = f(x) + \frac{\Delta x}{1!}\frac{\partial f(x)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \cdots$$

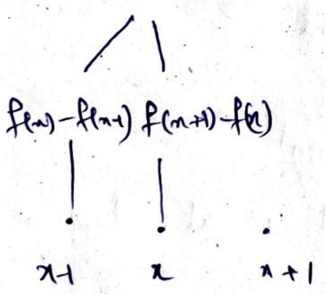
1-D

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

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

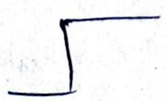
$$\frac{\partial f(x)}{\partial x} = f(x+1) - f(x)$$
$\hookrightarrow$ fwd diff

$$\frac{\partial f(x)}{\partial x} = f(x) - f(x-1)$$
$\hookrightarrow$ bwd diff



Ramp 

$\frac{\partial f(x)}{\partial x}$
- $\rightarrow$ zero for area of constant
- $\rightarrow$ Non zero at the onset and along the ramp

Step 

$\frac{\partial^2 f(x)}{\partial x^2}$
- $\rightarrow$ zero for area of constant
- $\rightarrow$ Non zero at start and end of ramp/step
- $\rightarrow$ along the ramp, it is zero

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

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

→ spatial kernel for 4-nbd

1	1	1
1	-8	1
1	1	1

→ 8 nbd $\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} + 2\frac{\partial^2 f(x,y)}{\partial x \partial y}$

$$\nabla^2(f(x,y)) = \text{Sum of nbd} - 4 \times f(x,y)$$

$$\frac{1}{4}\nabla^2(f(x,y)) = \underbrace{g_{b4}}_{\text{average of nbd}} - f(x,y)$$

$$\frac{1}{8} \text{ ---} = g_{b8} - f(x,y)$$

dark

(shows only the edges) ⇒ very dark ⇒ visually poor but contains information

⇓

(for visual improvement) Image Crispening

$f(x,y) = g_{b4} - f(x,y)$

Add -ve of laplacian to the original.

↓ Crispening

$$\hat{f}_c(x,y) = f(x,y) - g_{b4} + f(x,y)$$

$$= 2f(x,y) - g_{b4} = f(x,y) + (f(x,y) - g_{b4})$$

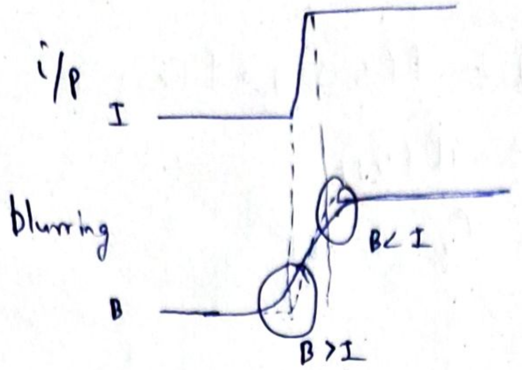
if $f(x,y)$ is more than g_{b4}

$f(x,y)$ becomes more

If $f(x,y)$ is less than g_{b4}

$f(x,y)$ becomes lesser.

Unmasking (for Sharpening)

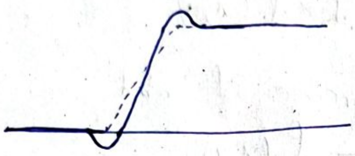


I – B

~~Mask~~ Mask

Mask + Input ⇒

Rich → Richer
poor → poorer



→ sharpening

Input Image is blurred & blurred ~~o/p~~ is subtracted from i/p to generate the mask. Add the mask to the i/p.

Homomorphic Filtering (Multiplicative Noise)

$\underbrace{I(x,y)}_{\text{Illuminated}} * \underbrace{R(x,y)}_{\text{Amount Reflected}} = f(x,y)$ (Intensity value at a point)

Amount Reflected } Reflective property.

Illuminated

⇓

Ideally same for all ~~random~~ points in the scene.

→ Actually small deviation is present.

(goal is to remove this variation)

Undesired

$\boxed{\text{Variation of } R(x,y) \text{ is Higher}}$

$\log f(x,y) = \log I(x,y) + \log R(x,y)$ → log is also ↑ fxn take DFT and apply High Pass Filter.

$$IFT\left(H(u,v)\ DFT(\log f(x,y)) \right)$$

$H(u,v) \rightarrow$ HPF

$\downarrow$ Exponential transf

$f(x,y) \cdot t = 1$

$\downarrow$

after doing all -1

$\downarrow$

for remaining log a

$f(x,y) \rightarrow \log \rightarrow DFT \rightarrow H(u,v)$

$\downarrow$

$\hat{f}(x,y) \leftarrow Exp \leftarrow IFT$

<u>Image Segmentation</u> → Extracting the Area of Interest.

Discontinuity Based

Boundaries b/w the regions. (discontinuity in terms of property) → edges, lines, isolated points.

Homogeneity based

Provides Regions where properties are similar.

• Different

$Prop(R_i \cup R_j) = false$

R_i and R_j are neighbouring Region

$I \Rightarrow \{R1, R2, - Rn\}$

• Completeness i.e $I = \cup R_i$

• Connectedness

all pixels in a Region is connected.

• disjointness

$R_i \cap R_j = \phi \ (i \neq j)$

• Homogeneity

$Prop(R_i) = true$.

edges ⇒ substantial change in intensity gives rise to edge pixels. Connect edge pixels, form the edge

lines ⇒ On both sides of it, intensity values are either low/high.

Isolated points → A background (foreground) surrounded by foreground (background) pixels.