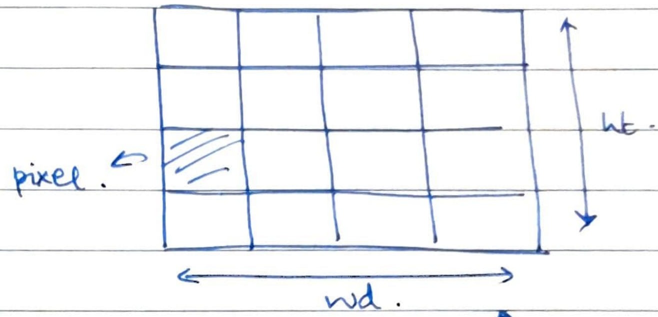
11/08/20.

Digital Image Processing.

Digital Images are an array of numbers representing colour intensities.

Uses 8 bits to rep. a pixel's intensity.



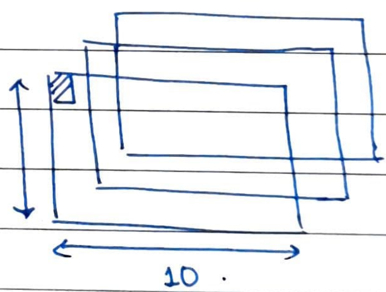
0 ⟵⟶ 255.
black   white.

Grey scale   8 bits / pixel & 1 2D channel.

→ Images are rendered from the video card reader.

→ More common : RGB notation / representation.

3 - 2D channels.
24 bits / pixel
∵ 8 + 8 + 8 [for 3 channel].



In fMRI :-
56 2D channels used.

- Classifying Images:

→ Via source :-

a) EM spectrum :-
  ~ γ rays : PET scans.
  ~ x-rays : greyscale ⇒ surveillance use colour changing techniques.
  ~ ultraviolet : currency forging dipped in material.

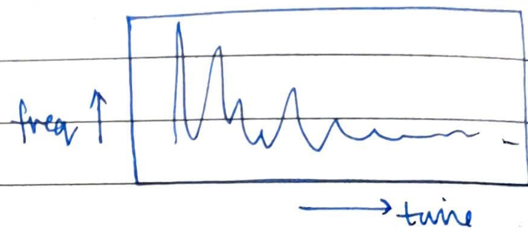
most dominant way of gen images

- ~ visible spectrum.
- ~ Microwaves :- can help detect images by avoiding obstructions ie: clouds.
- ~ Radio waves : fMRI.

b) Ultrasound :-

- sonography : bouncing back of waves based on distance.
- Spectogram : based on ~~x~~ audio freq/ intensities generate graphs.

For different words being said, different plots can be obtained.



c) Electronics :-

- Electron beams shot on surfaces (focused) to get a microscopic view.
- The changes in the properties of the beam is what turns into an image.

d) Computer generated :-

→ Reflection images : commonly formed & perceived by human eyes.

→ Absorption images :- primarily about internal structure (just ~~outer~~ structure) bone damages etc.

→ Emission images : primarily about internal properties tissues cartiladge etc.

classmate
Date
Page

Eg. Barium milkshake : emits radiations . & using this diff kinds of analysis info is deduced.

X- Rays are opposite in this sense ∵ here the source lies outside the body.

→ Enhancement : highlighting certain properties of images.

→ 3D images : Stereo images
↳ use specialized cameras to mimic the human eyes.
↳ Given a L & R pair we can figure out the depth of the image.

→ Multiview images : capturing a 3D image in its entirety.
Diff b/w multiview & stereo ⇒ can view with complete details in multi.

x———————x

Processing :
→ creating newer images } algos.
→ retrieve its attributes }

→ consumer based views.
↳ making analysis.
↳ raw human eyes.
↳ consumption by machine based processes.

I <u>First half of course</u> :-

→ work on enhancement.
→ restoration.
→ enhance colour images.
→ Morphological proc.

II : <u>Second half</u> :

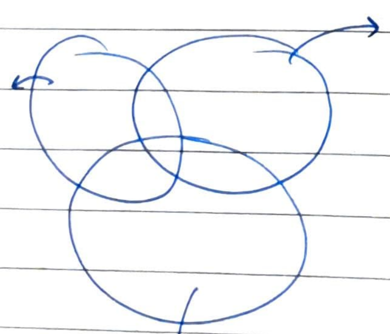
→ Segmentation : define groups
→ Desc: roundedness, elongation
→ Object recog.

] work on image attributes

<u>Misc</u>:

- Contrast detection.
- feature detec (panaroma.
→ B/W to colour algorithms.
→ Image restoration

DIP.
pixel → pixel



CV
pixel → symbol.

graphics : symbol → pixel.

→ Libraries :- scikit - image
opencv-python
pandas
scikit-learn.

12/08/20 : Tutorial :

→ select my roll number 2018101041
→ proceed like a normal repo after that.

→ keep commiting & pushing code ∵ remote repo!

Configure : git config --global user.name <username>
" " --global user.email <email ID>

Colab :-
- run bash commands using !
- pip install opencv-contrib python.

14/08/20

TA hours :- $\boxed{\text{Saturday 4:00 - 4:45}}$.

→ Consumers of an image processing algo :- Humans.

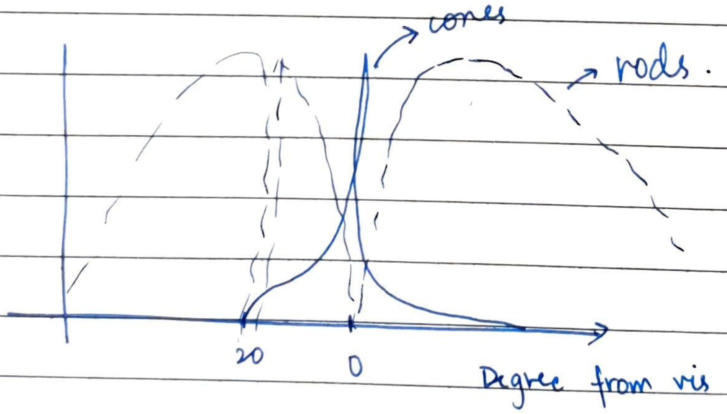
- $\boxed{\text{Human eye:}}$
  - Ball of diameter 20mm.
  - Light enters the eye, passes through the lens
  - Chamber filled with liquid vitrious humor.
  - Cornea shrinks & expands based on amount of light
  - Lens absorbs IR, UV rays (capability dec. with age)

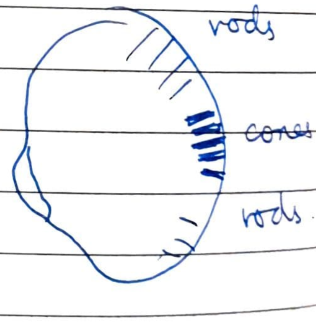
Image formation occurs on the retina.
Light receptors : Cones & rods.

- located in fovea region.
- respond to bright light.

→ Rods - generally become active at low-~~minimum~~ illumination regions : scotopic vision.

no: of rods/ cones.





→ Fovea considered to be a square sensor array (1.5mm x 1.5mm)

Eye lens is flexible (variable focal length). with the help of ciliary muscles. (14 - 17 mm).

→ The unit Lambert is used to measure intensity.
$(10^{-6} - 10^{4})$ range. But this range cannot be perceived simultaneously.

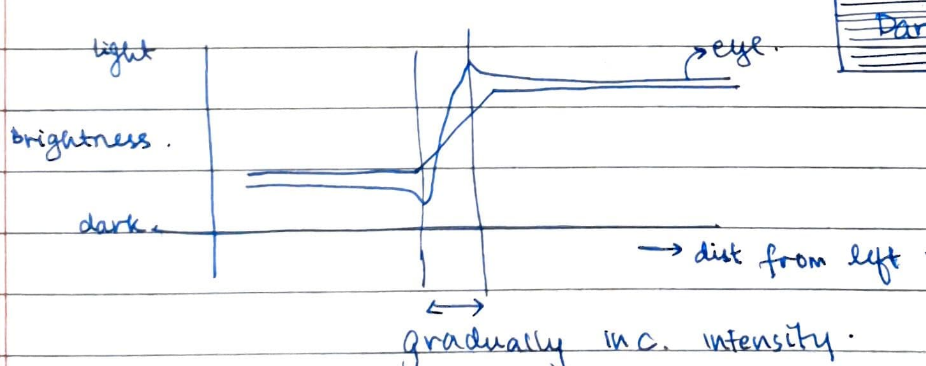
Extent to which subjective brightness (→ brightness perceived.) can be perceived → adaptivon range.

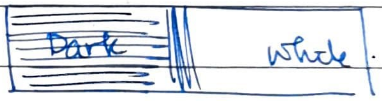
Brightness disc:
For lower intensities human eye is very sensitive.

ie: $\frac{\Delta I_c}{I}$ → small weber ration ⇒ good disc.
larger weber ration ⇒ poor disc.

→ Psychovisual Effects:

- Change in 10% intensity may not be perceived as 10%.
- Mach band:



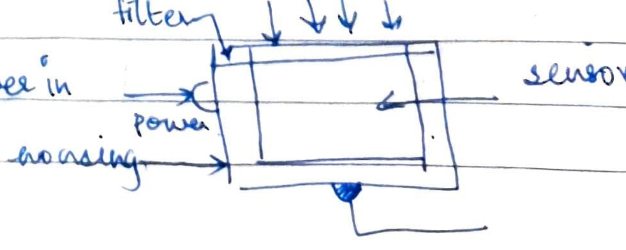


contrast?

- **Digital Image Acquisition ::**

Principle :- convert energy to intensity (numeric).

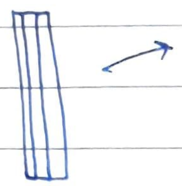
When no energy, volt out = power in



→ Scanners :-
Light is shined on the doc; → reflected to a series of mirrors → passes through ~~atten~~ a lens → converted to voltage → acquire image.

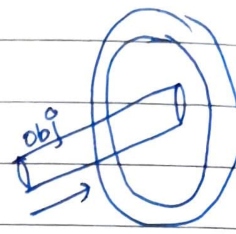
ADC
Analog to digital converter

→ <u>Linear array acquisition</u> :-



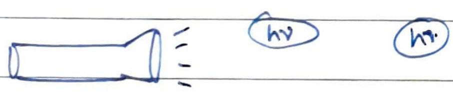
→ single single rows.

→ <u>3D image</u>



Each portion acquires a cross section.
Integrate all to obtain image.

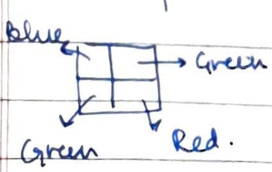
• Light can also be particle (photon) nature.



it is the motion of ē that produces the electricity & change in voltage. [Used to exploit & convert light energy to voltage].

Light sensing diodes are sensitive to 'light' but not to 'colour'.

→ To capture colour, use the <u>Bayer Filters</u>



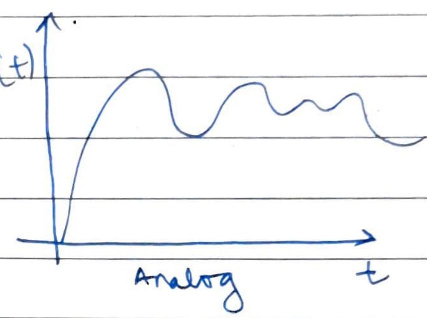
| using same sensor array should work to reduce area & ~~cost~~.

Relatively more ~~Set~~ green sensors → human eye sensitive to shades of green.

→ $\boxed{\text{Demosaicing:}}$ process of getting colour from ind. grey scale & ind. colours.
→ borrow colour from neighbours.

- **Image Sampling & Quantization:**

Images are essentially signals.

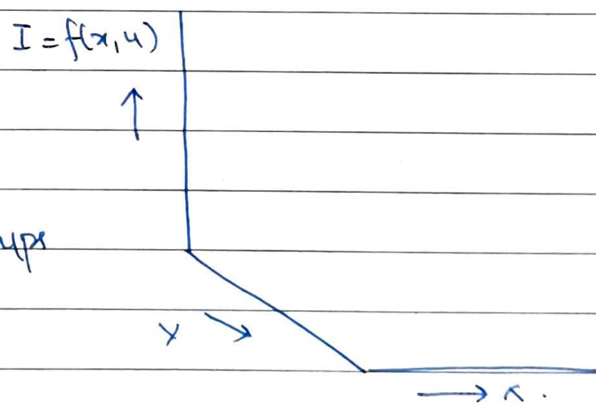


Notion:

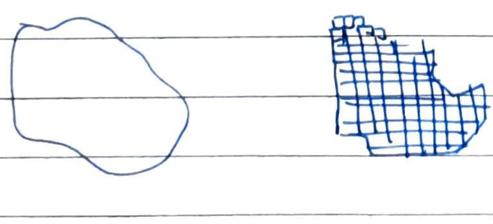
Images are a function of discretized space.

Intensity at a particular point is a $f(x, y)$.

Forms a neat 3-D space.



The signal is captured at certain quantized timestamps ie: sampling.



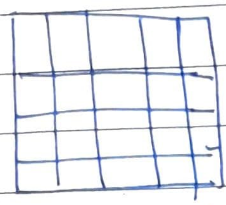
→ What is a resolution of a sensor?

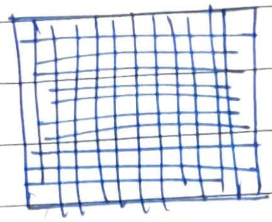
A: The total no: of photosites (however some photosites are dead or useless).

Size of the sensor plays a huge role: Graph paper.

Sampling :- measuring the frequencies at regular intervals of time.
Quantization : Expressing thes freq. as quanta.

→ Resolution isn't the only deciding factor
: A pixel's size is also important.





coarser look better.

↓

pixel pixel.

→ Aspect Ratios :

Ratio of $\frac{\text{width}}{\text{height}}$ = AR.

→ Spatial quantization :-
variation in no: of sensors $(x \times x)$ → sampling.
Tentensity quatitazation :- intensities merge.
(colour distorts) 8 bits/pixel → 4 bits/pixel etc

18/08/20

Image : a function defined over a discrete dom

ie: $f(x,y) = z$.
↓ Domain: (x,y) ↓ Range. (Intensity)

→ Sampling :- quantization in spatial domain
values only in certain spatial quant.
→ Range sampling : 8 bits/ pixel 2 etc.

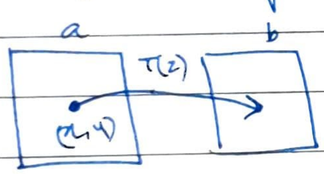
→ Other factors: Temporal Sampling.
aperture shutter speed ISO.

Smoothened version of ocean ⇒ got by shutter speed.

- **2 Paradigms in image processing:**

→ Spatial domain: manipulate pixels in SP.
→ Transform domain.

→ Intensity Transform: **Point to Point** :-



$a(x,y) = z$.

$b(x,y) = z' = T(z) = T(a(x,y))$.

Attacks ~~the~~ intensity.

Eg: converting to negative of the image to clearly observe / focus some areas.

Img 1: r. $T(r) = s = (L-1) - r$.   ↗ in neg image.

→ Brightening / Darkening image.

Dark :- $T(z) = z - k$ [red. intensity]
Bright :- $T(Z) = z + k$ [inc. intensity]
} ensure $T(z)$ remains within domain.

If at all it exceeds: do clamping: set equal to end values.

→ Scaling to colour map range. (linear mapping).
For eg: 4 bit/pixel $0 \to 0$ $15 \to 255$
Doing so ⇒ we're also increasing no: of bits.

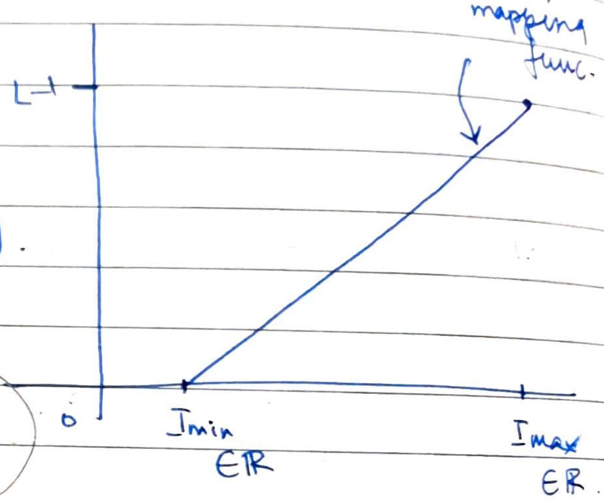
So far all transforms are <u>Linear</u> intensity transforms

$$T(z) = K_1 z + K_2.$$

∴ Mathematically :-

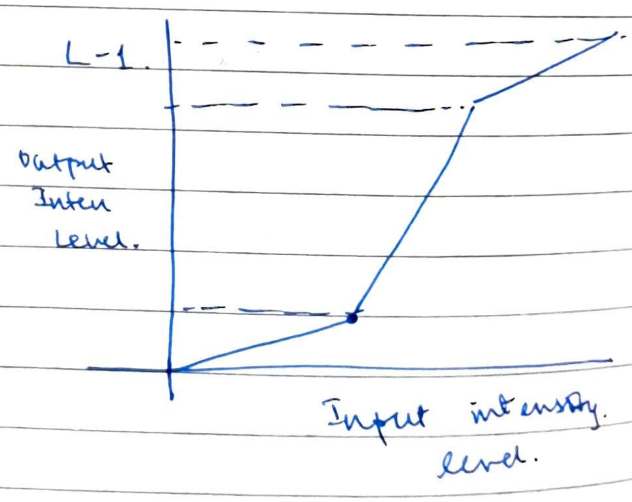
Normalize

$$J = \text{round}\left(255 * \left(\frac{I - \min(I)}{\max(I) - \min(I)}\right)\right).$$

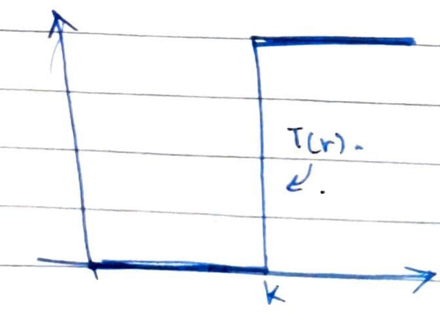


→ Piecewise Linear Transform :

Extreme form of this kind of piece wise function is called THRESHOLDING.



if (Inp I) < K ⇒ set 0
else ⇒ set 255.



(r_1, s_1) $(K, 0)$
(r_2, s_2) $(K, 255)$

Non-Linear Intensity Transformation :

- Too many values.
- Range is very large eg: 0 - 255.

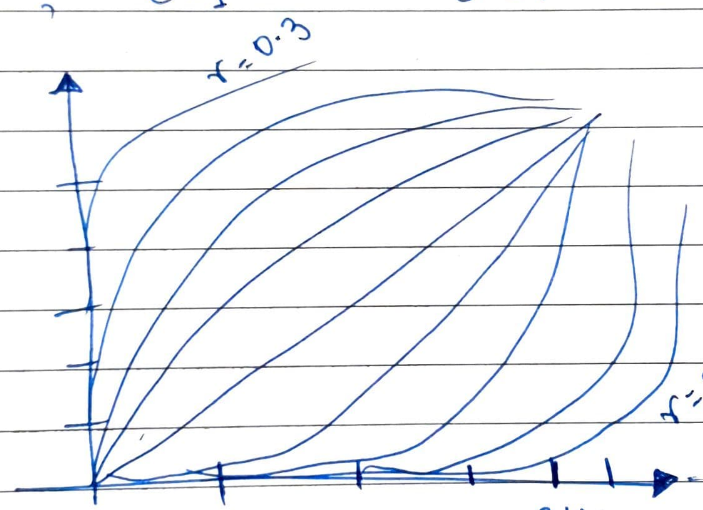
TRICK → Use log transformations :-

$$T(r) = c\log(1+r). \quad [\text{Non Linear}]$$

→ Power Law transformations :-

$$s = cr^{\gamma}$$

if $\gamma = 1$, $c = 1$ $\quad s = r$ .... copy transformatio



344.

brighter portions get mapped vv. well.

→ Larger no: of white intensities get compressed.

→ One kind of haze removal can be done in this way

Intensity slicing :-

21/08/20

LECTURE 4:

3 approaches to spatial domain processing
→ point to point :-
- linear intensity.
- piece wise : thresholding.
- power law transform.

Bit Plane Slicing:

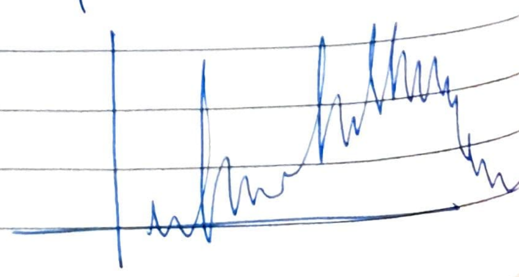
Each pixel represented by 8 bits.
- collect least significant bits ⇒ you'll get bitplane-1.

- We observe that 8^{th}-bit plane is a close approximation to the original image.
[Revisit topic ⇒ image compression].

- uint8 for numpy will be stored.

- Histogram: visualization + representation.

$h_r(i) = n_i$
i = intensity values.
n_i = no: of pixels with intensities i.



We can view histograms & gauge the intensities or contrast in the img

$$\text{contrast} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \in [0, 1]$$

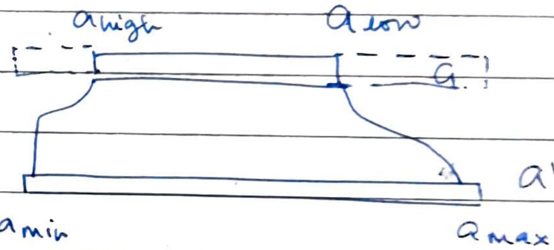
→ After image processing we can look at the histograms & then predict whether or not the proc was good.

→ Higher contrast ⇒ spread across intensities

☆ reconcile image & intensity distribution.

- Contrast Stretching: $y = (\quad)(x) + c$

$$f_{ac}(a) = a_{min} + (a - a_{low}) \cdot \frac{(a_{max} - a_{min})}{(a_{high} - a_{low})}$$



↓

histogram thus gets stretched or spread apart.
ie: ~~spread~~ of line stretching :

→ If single pixel with 255: stretching happens only at left

→ If single pixel with 0 stretching happens only at right

→ Similarly : if 1 pixel 0 & 1 pixel 255
✗ no contrast stretching.

- Another method :-

$$a_{low} = \{ i \mid H(i) \geq M.N (s_{low}) \}$$

$$a_{high} = \{ i \mid H(i) \leq M.N (1 - s_{high}) \}.$$

→ some % (3%, 5%) using this selected

- Intensity aware operation:
  → Histogram equalization.

Intention Transform an image such that the histogram of the transformed image is equalized

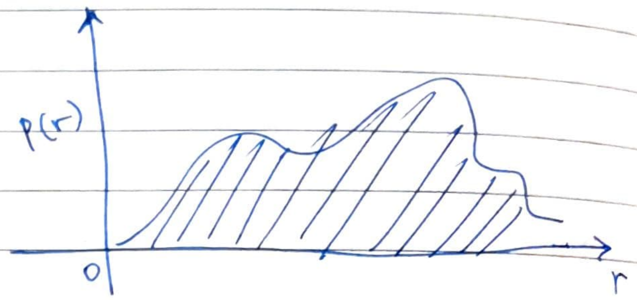
→ normalized histogram → probability dist.

→ For the purpose of analysis, we can consider continuous distribution.

Consider R be the R.V for the intensities

$$\int_{-\infty}^{\infty} P_R(r) = 1.$$

Practically :- $\int_{0}^{255} P_R(r) = 1.$

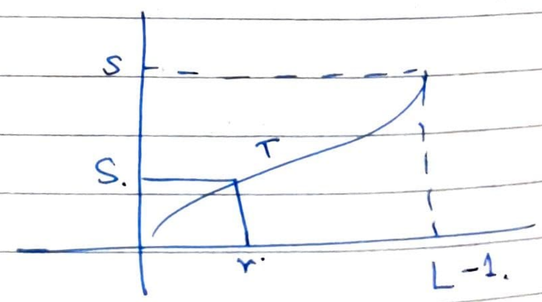


→ Let R be the R.V for the image. Consider another R.V S.
We want a uniform dist. in S.

Need a transformation T ST: $s = T(r)$.

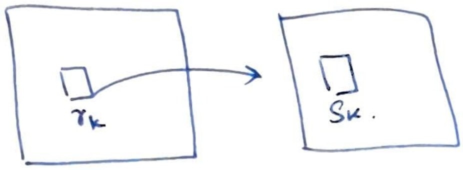
For S to have uniform dist, we should have the

$$P_S(s) = \frac{1}{L-1}.$$



- T is usually prefered to be monotonic.
  2 R.V related by a monotonic transformation then

$$\boxed{P_S(s) = P_r(r)\frac{dr}{ds}}$$ → rate at which it changes WRT target



ie: $p_s(s)\,ds = p_r(r)\,dr$.

Sub for $p_s(s)$:-

$$\frac{1}{L-1}\,ds = p_R(r)\,dr.$$

$$ds = (L-1)\,p_R(r)\,dr.$$

Int both sides:-

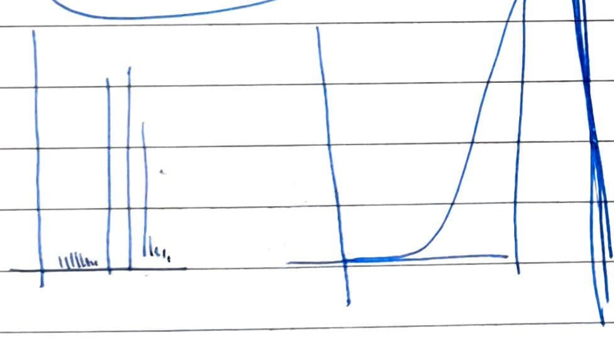
$$\int ds = \int (L-1)\,p_R(r)\,dr.$$

$$s = (L-1)\int_0^r p_R(w)\,dr.$$

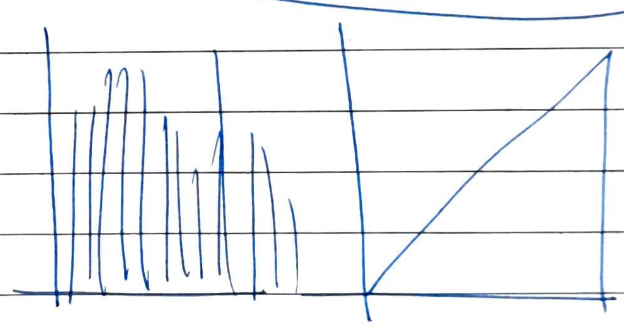
$$\boxed{s = T(r) = (L-1)\int_0^r \underbrace{p_R(w)\,dw}}$$

use the CDF to subst. & compute.

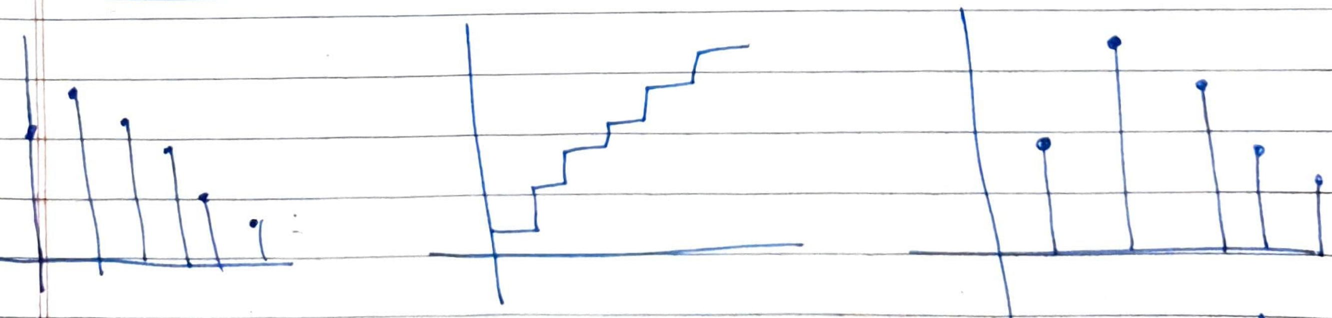
That is: Original :-



Histogram equalized



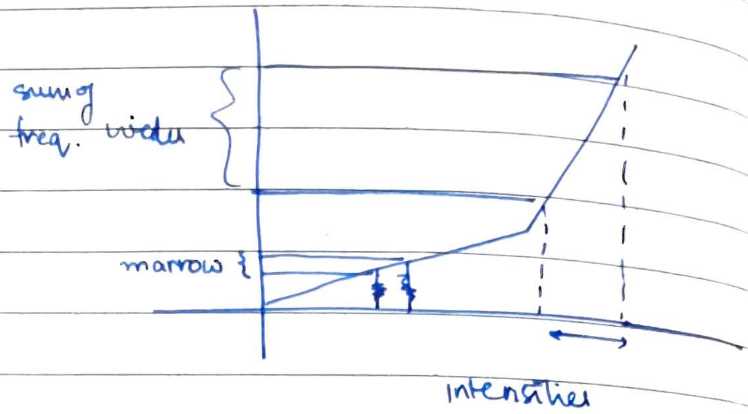
- Histogram equalization. → enhancement procedure.



$s_k = \text{round}\left[(L-1)\sum_{j=0}^{k} p_r(r_k)\right]$

Equalized.

When a ~~lot of intens~~ narrow band of intensities maps to a wider band $\Rightarrow$ low contrast.



→ Contrast enhancement & histogram equalization may differ from situation to situation.

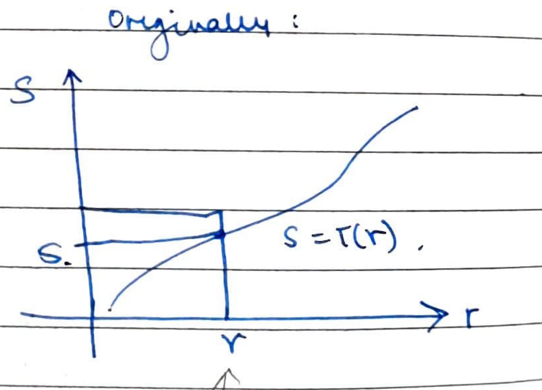
range of contrast & histogram are roughly same

• <u>Histogram Specification</u> :

Prob. dist over S

$P_S(S) \longrightarrow$ uniform.

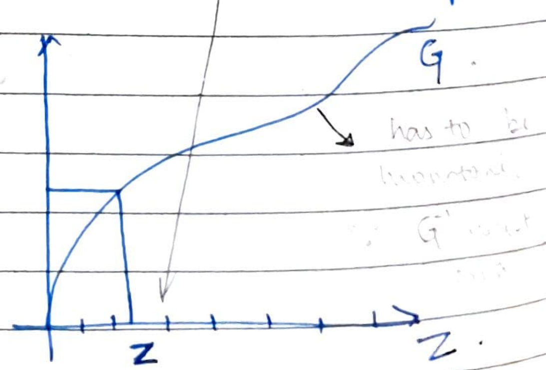
& T is the transfer func.



Let us have a new transfer function that we define.

– Specify a custom curve G.
$\exists$ some value of z that maps to s

ie: $s = G(z)$ CDF of reference image



Consider strictly monotonic increasing.

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

GIMP tool for image processing.

Now, we have z as a function of r.

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

ie: $z = U(r)$.

Histogram specification can have a curve from a reference image.

→ The output image would then have the same intensity style & intensity profile as the ref image.

Histogram Equalization is an example of global to point processing.

→ Histograms:

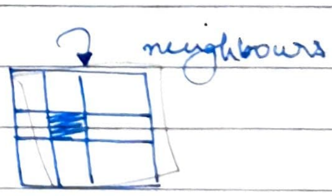
- More of a visualizations.
- useful statistic of image intensities.
- Not dependent on image size.

Dis:-

- No spatial info.
- Intensity centric

• Neighbourhood To Point:

Neighbourhood based transition of this process.



Map a group of say x by x pixels to the same n by n pixel group.

↳ manipulate this value & proceed.

Conditional Image Enhancement :

$P(r_i) = n_i/n$.

$m(r) = \sum_{i=0}^{L-1} p(r_i)\, r_i$.

$\sigma^2(r) = \sum_{i=0}^{L-1} (r_i - m)^2\, p(r_i)$.

} Stat params.

Objective : Enhance dark image areas leaving light areas unchanged.

step 1 → Identify 'dark area pixels' & 'light pixels'.
some special proc to check.

step 2: → Enhance dark area pixels.

1. Using some kind of an if condition check all pixels.
→ use some $\boxed{m_{s_{xy}} \le k_1\, m(r)}$ helps ~~for~~ find candidate pixels

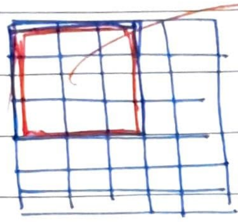
Here 'enhance' means contrast enhance.

$$\boxed{\sigma_{S(x,y)} \le k_2\, \sigma(r)} \rightarrow k_3\, \sigma(r) \le \sigma_{s_{xy}} \le k_2\, \sigma(r).$$

↳ < 1.

28/08/20

- Spatial Domain Filtering.
  (Intensity transform eg.)



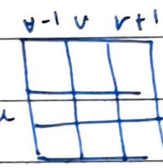
Take mean of all 9 values & replace with avg of all.
round (avg all neighbours)

High difference of intensities will be replaced by closer intensities.

→ Weight mask, kernel, filter.
Mask. Co-efficient sums to = 1.

ie:

$$H \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$



Transform :-

$$I' = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(u+i, v+i) \cdot \underbrace{H(i,j)}_{1+i,\ j+1}.$$

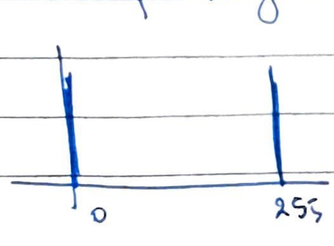
→ With a larger averaging filter size :-
3x3 → 5x5 → 7x7 blurrer & blurrer

Image enhancement ?

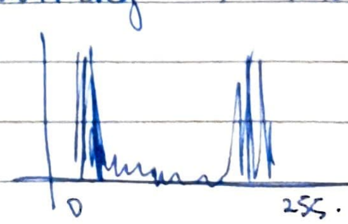
For most natural photographs ⇒ blurring ∵ more smaller values.

Smoothening : Forms first step in many image proc pipelines

→ Binary image ⇒ blurring ⇒ intensities spread apart.



⇒



3x3

. . .

1) Visually you can get the effect of larger filters by smoothening repeatedly with smaller filters.

2) Op. with smaller filters would be faster as well.

- Weighted averaging:

eg: req. to remove undesirable probabilities.

1	2	1
2	4	2
1	2	1

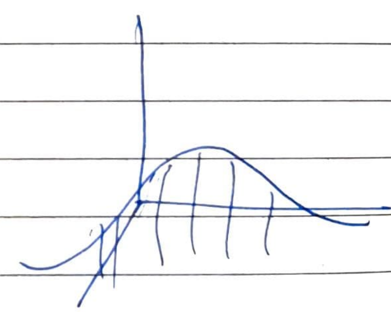
higher weighted

Giving central pixel more value ⇒ gaussian filters.

Gaussian function :- param μ, σ.

$$F(\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Sample a gaussian.



Specify the (s, neighbourhood)

Pick a value for filter size such that it covers a good range of the image.

As $\sigma \uparrow$, blurring $\uparrow$.

$\therefore$, if σ is small, the center pixel given max wt.

- Derivatives:-

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \qquad f'(x)\big|_{x=a}.$$

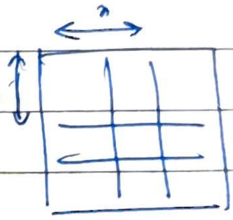
2D:-

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}.$$

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta y \to 0} \frac{f(x, y+\Delta y) - f(x,y)}{\Delta y}.$$

Digital first approximation:-

First $\quad \dfrac{\partial f(x,y)}{\partial x} \approx f[x+1, y] - f[x,y].$



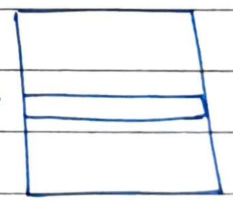
Second Dr $\quad \dfrac{\partial^2 f(x,y)}{\partial x^2} \approx \big(f[x+1,y] - f[x,y]\big) - \big(f[x,y] - f[x-1,y]\big).$

This is essentially ~~the~~ a 2×1 rectangular filter.

-1	1.

center

Run the filter across the row here
The resultant image will then
contain the first derivatives.



Rapid change in intensity : corresponds to an edge.

1st derivate : tells us where there are edges.
2nd derivative : tells us the location of edges.

Symmetric Diff. : $\quad f'(x) = \lim_{h \to 0} \dfrac{f(x+h) - f(x-h)}{2h}.$

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

y: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

- Gradient in 'x' tells where all intensity is changing as you go from L – R.

- Likewise same thing for top – bottom also.

Can ∴, record the presence of images.

- Prewitt edge filter.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge $\perp$ to gradient.

classmate

Date
Page

1/09/20

- **Recap:**

- Mean/Avg filter.
- Operation : specify a mask : neighbourhood & then entries
- Filter entries $\Sigma = 1$.

Mask size: perform image smoothing with larger filter size - blurring increases.

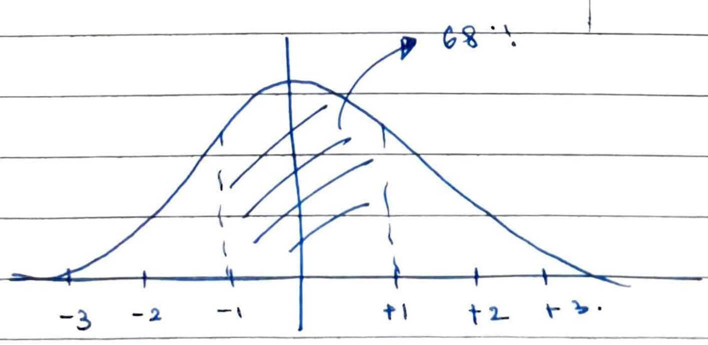
Repeated averaging : increased blurring.
↳ used to approx gaussian filter by using smaller set of cheaper filters.

→ Gaussian filter:
- arriving at the Gaussian ... how?

PDF : $g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}$.

$\int g_\sigma(x)\, dx = 1.$

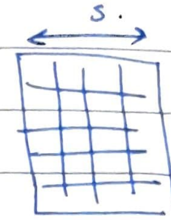
→ Taking 1 unit σ : 68%. | 3σ: 99.7%



If you take upto 3σ : you'll get the filter more or less.

→ How to decide the size of the filter.

To select ~~#~~ S



$$S = 3\sigma + 3\sigma + 1\sigma$$

$$S = 7\sigma$$

This would represent the filter more or less faithfully.

To avoid strange boundary effects - take odd sized filters.

→ Gaussian's co-efficient can be obtained via the Pingala / Pascals triangle.

Σ coeff for $N = k$: 2^k.

For 1D: For. eg $S = 7 \times 7$. ⇒ select row 7.

$$\frac{1}{64} \begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix} = V^T.$$

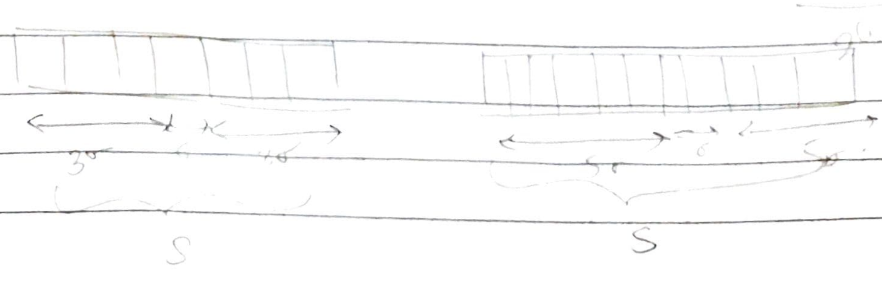
For 2D filter, take outer product of $V \times V^T$.

ie: $\frac{1}{64} \begin{bmatrix} v \end{bmatrix} \times \frac{1}{64} \begin{bmatrix} v \end{bmatrix}^T$.

Hence we can obtain the digital. approx to gaussian filter given a value of σ..

Peak / central val of gaussian is got by setting $x =$ [illegible]

$$\frac{1}{\sqrt{2\pi}\,\sigma} = \frac{{}^{N}C_{N/2}}{2^N}. \qquad \boxed{N = S - 1}$$



- **Edge detection:**
  - Area in our visual cortex dedicated to see intensity variation.
  - Edges are a key IP op.
  - sudden changes/ discontinuities to intensities.

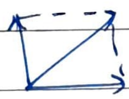
Obtain greater support for edge we use prewitt edge filters.

Edge is $\perp$ to gradient / intensity change.

→ Given an image

Obtain the $\frac{\partial f}{\partial x}$ $\frac{\partial f}{\partial y}$ image.

At any pt.



magnitude & orientation info.

loc. [edges are strongest] [captures direction.... gen. manner in which grad changing]

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- **2 gradients:**

< >

For a functions: sum of its derivatives = Laplacian.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

Laplacian op. $\nabla^2 f = f(x+1, y) +$

0	1	0
1	-4	1
0	1	0

$\Sigma = 0$

SOBEL

~ Sobel X : filter :
Sweep from L-R. finds intensity changes vertical edges

~ Sobel Y : filter :-
Intensities in vertical direction.

Unlike edge filter $\neq$ sum to 1. sum to 0 ✓

These filters are symmetric : to avoid giving preference to any 1 particular direction.

Sobel is not sensitive, & so is less noisy.

- Gradient Filters → Image Enhancements.
  Image Sharpening

$I(u,v)$ get $\nabla^2 I(u,v)$

Add to old image: $\nabla^2 I(u,v)$ ~~I 128~~ + $I(u,v)$.
Result → Sharper.
Only vis $\nabla^2 I(u,v) + 128$.

- High Boost Filtering :-

what does blurring take away.

ie: $\boxed{\text{original}} - \boxed{\text{smoothened } 5\times5} = \boxed{\text{detail (Edge info)}}$

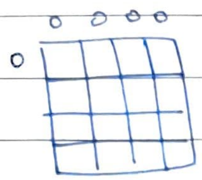
$\boxed{\text{original}} + \underbrace{a}_{\text{Scalar } a} \boxed{\text{details}} = \boxed{\text{Sharpened}}$

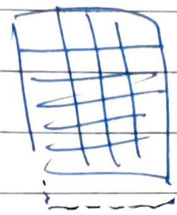
Can scale the laplacian also & attain the sharpening

∇^2 : unsharp masking

blurr method : highboost filtering.

- Corner cases & Padding :

→ 



replicate border entries.

pad with 0

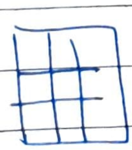
Do not cause much issue unless object of interest is on the boundary.

Advantage: 5x5 image $\xrightarrow{\text{filter}}$ 5x5 image.

valid filtering 5x5 → 3x3.

SO FAR ALL LINEAR FILTERS.

- Non-Linear Spatial filters:

→ Max Filter :  max of ∈ 

Can be used to get rid of pepper noise.

→ Min filter

can be used to get rid of salt noise.

Tend to become little dull.

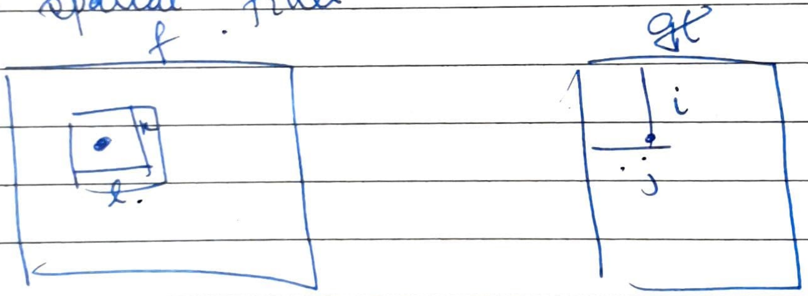
Median filter : 50% filter. can get rid of salt & pepper noise. adaptive.

! Mean extremely prone to outliers.

All rank order filters.

- Edge preserving smothing : BILATERAL FILTER.
  Record posⁿ where edges present
  ↳ don't ~~p~~ do smoothening in those areas.

Linear spatial filter :



$$g(i,j) = \frac{\sum_{k,l} f(k,l)\, d(i,j,k,l)}{\sum_{k,l} d(i,j,k,l)}.$$

4/07/20

Binomial image

- **Recap:**

faster

Mean: blurs, removes simple noise, x details

Median: preserves some details, removing strong noise

Gaussian: blurs, preserves details only for ↓ σ.

better than mean

→ **Edge Preserving Filtering:**

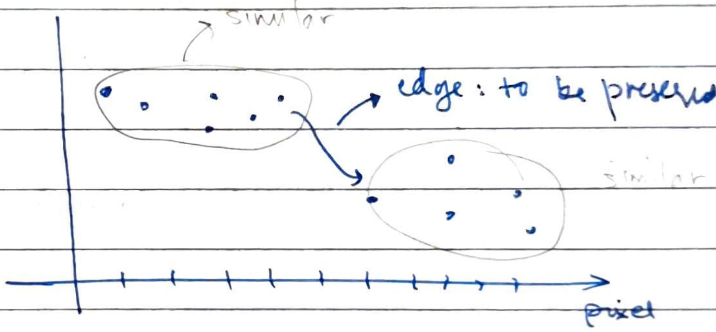
Smooth isotropically: only at edges:-

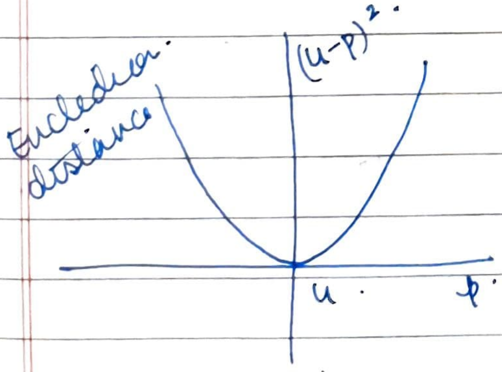
Eg: Consider 1 row:-

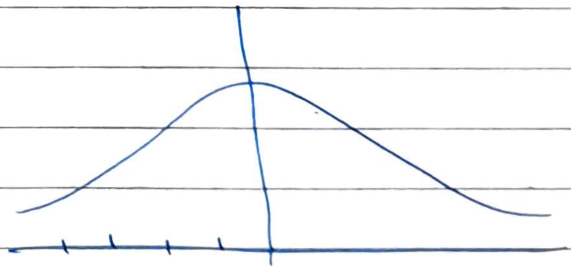
Now consider the gaussian

$$W_c(p) = \exp\left(-\frac{(u-p)^2}{2\sigma_c^2}\right).$$

rate at which it falls







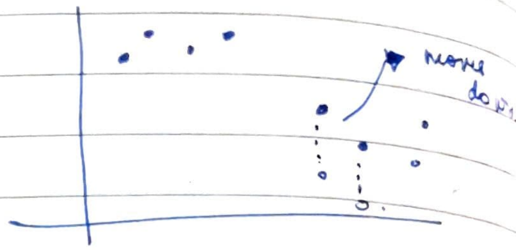
The neighbourhood values pull down the image.
∴, the two edge diff. values come closer & closer.
The contribution to sum by wts is diff. as distance increases from u.

since with intensities : called photometric weights.

As a result, introduce another weight :-

$$W_c(p) = \exp\left(-\frac{(I(u) - I(p))^2}{2\sigma_s^2}\right).$$

Look at intensities also now:
For all intensities, they're
pulled down.



For similar intensities, wts ≈ 1.

This helps us preserve the edges.

$$W_{bi}(P) = W_s(P) \times W_c(P).$$

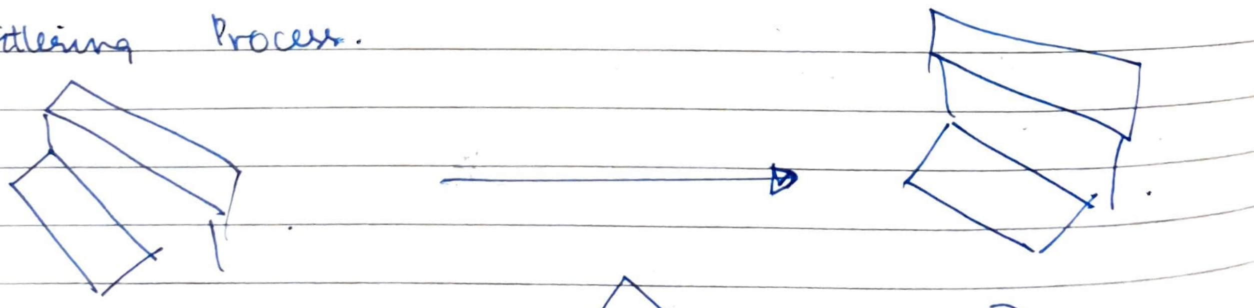
Denoise

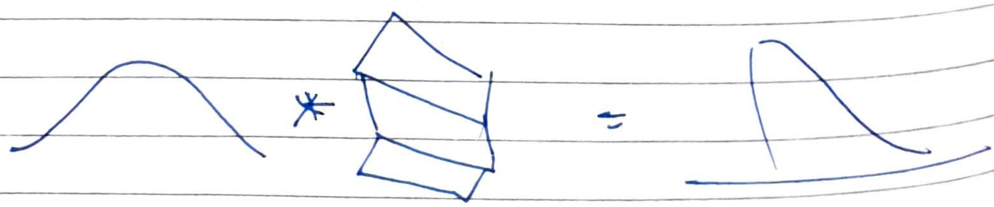
feature preserving

ie:

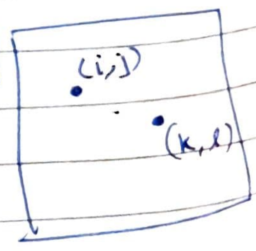
$$I'(u) = \frac{\sum_{p \in N(u)} e^{\frac{-\|u-p\|^2}{2\sigma_c^2}} e^{\frac{-|I(u)-I(p)|}{2\sigma_s^2}} (I(p))}{\sum_{p \in N(u)} e^{\frac{-\|u-p\|^2}{2\sigma_c^2}} e^{-\frac{|I(u)-I(p)|}{2\sigma_s^2}}}.$$

→ Filtering Process.





→ Same extension to 2D also: [domain kernel]



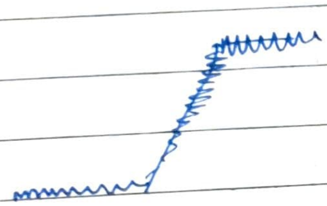
$$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right).$$

$$r(i,j,k,l) = \exp\left(-\frac{\| f(i,j) - f(k,l) \|^2}{2\sigma_r^2}\right).$$

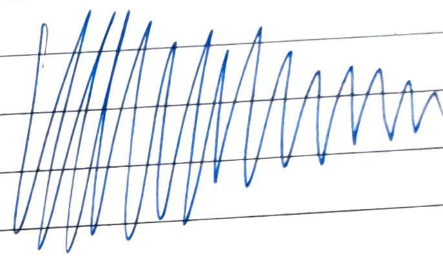
- data dependent.
- range filter
- kernel depends on image content.

Progressively decreases the difference b/w regions with similar intensities.

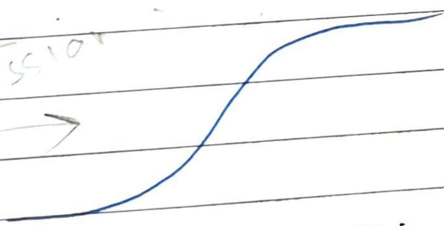
→ Effect of noise :



$\xrightarrow{\frac{df(x)}{dx}}$

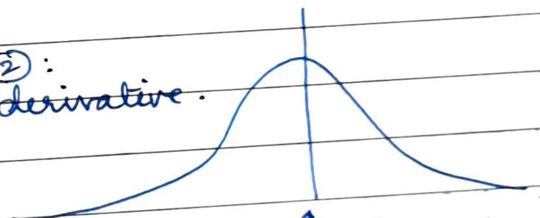


→ Solution : smooth first :



STEP ① : smoothing.

→ STEP ② : derivative.



↑ edge.

To save one step :- ~~In~~ since we're working with linear filters :-

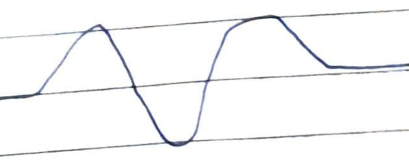
$$\frac{d}{dx}(f * g) = f \cdot \frac{d}{dx}(g).$$

$\boxed{g'(x)}$

Use the kernel as $\frac{d}{dx}(g)$

Further for $g''(x)$: Laplacian of gaussian.

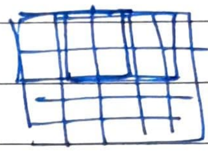
↳ Noise suppression.



Approx LoG with DoG → difference of 2 gaussian diff scale
Choose σ_1, σ_2 appropriately $\nabla^2 G \approx G_1 - G_2$.

- All the previous operations were called as linear filters since :-

Scaling :- $T(x) = y$; $T(\alpha x) = \alpha T(x) = \alpha y$.
Addivity :- $T(x_1 + x_2) = T(x_1) + T(x_2) = y_1 + y_2$.

Operation of shift & slide :-  : corellation

If we horizontally & vertically flip the filter value ⇒ convolution

~ Whatever we have seen uptil now are all directly manipulative pixel in spatial domain.

- TRANSFORM :

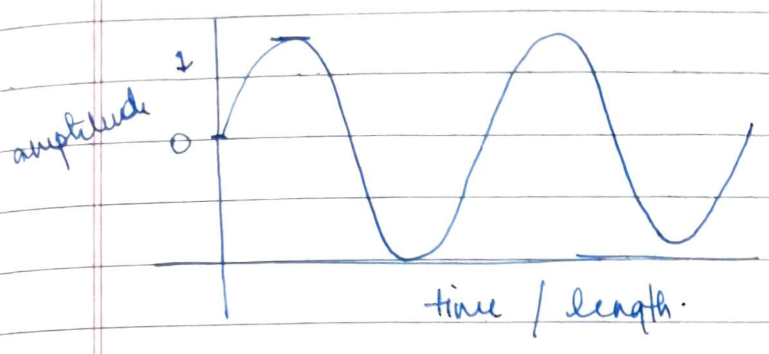
Input → Transform → Processing → Inv. Tran → Out.

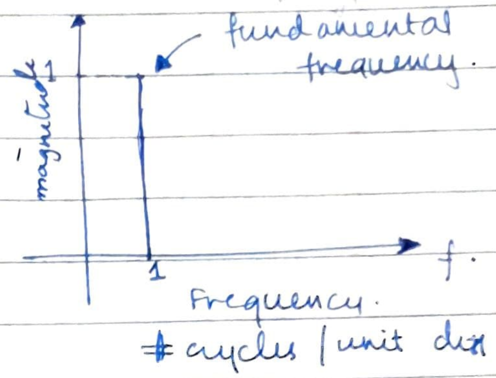
→ Periodic signals :-
$x(t) = A\cos(\omega t)$

Angular freq :- $\omega = 2\pi f$ radians / s.
$= \frac{2\pi}{T}$ T : Time period

A cos t

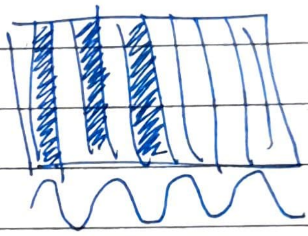
Image can be considered as signals.





spatial domain : How much diff to travel befor signal repeats.

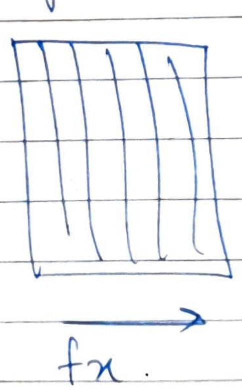
Eg:- $I(x,y) = 128 \sin(2\pi x/16)$. 128×128 spatial signal

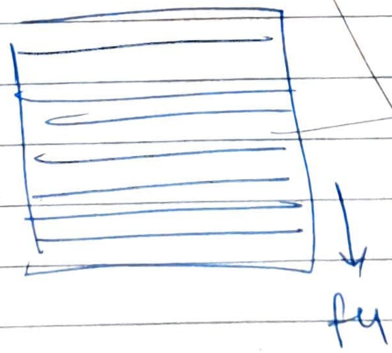


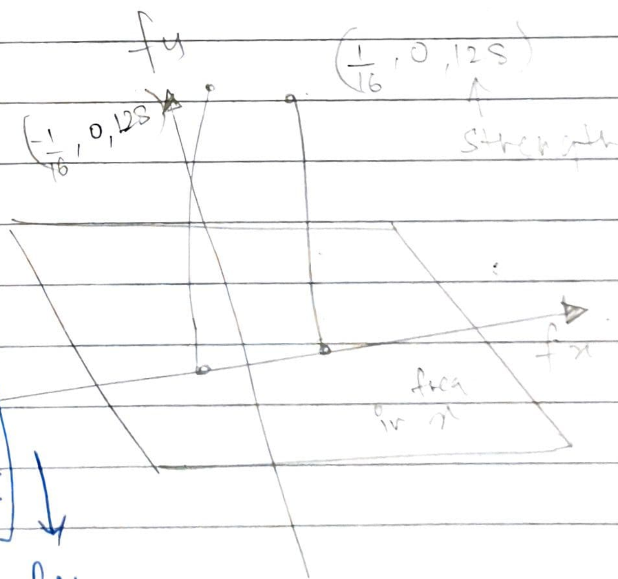
Smallest period we can have is $\boxed{\frac{1}{2}}$: 2 pixels spatial period.

If $f = \frac{1}{16} \Rightarrow$ 16 pixels के बाद repeat

The direction of the wave indicates direction of frequency.







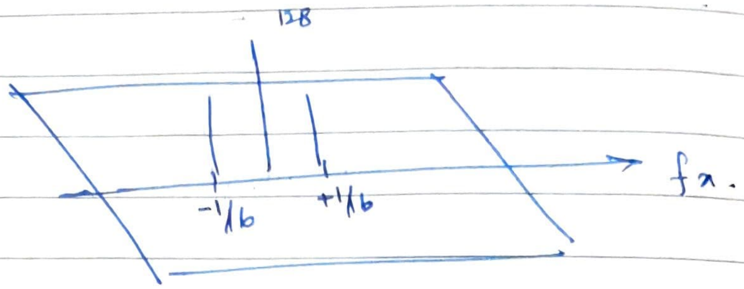
Note. as f ↓ pts. on f_x & f_y come closer

ie: if $f_x = \frac{1}{16}$ if $f_x = \frac{1}{4}$

08/09/20

Combination of wave patterns.

$$128 + 64 \sin\left(\frac{2\pi x}{16}\right).$$

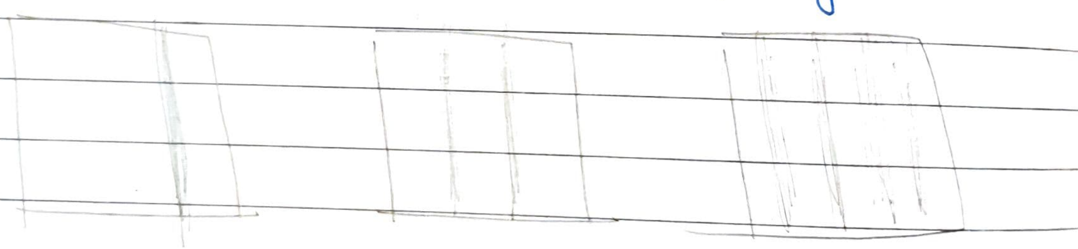


Consider :- intensity images for :

$$s(x, y) = \sin\left[2\pi(u_0 x + v_0 y)\right]$$

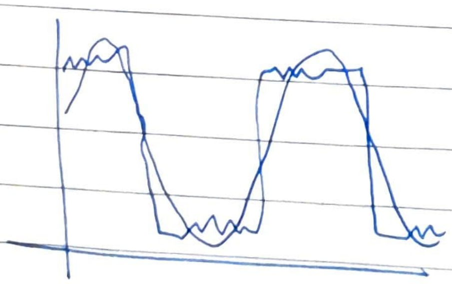
u_0 : freq in x v_0 : freq in y.

As V_0 ↑ then bend clockwise increases



- Fourier series : approximate periodic signals with sines & cosines.

Eg.



sin & cosine harmonics

$$y(t) = \sum_n a_n \sin(nf \times 2\pi t) + \sum_n b_n \cos(nf \times 2\pi t).$$

$$\boxed{\omega = \frac{2\pi}{T}}$$

$a \sin(n \omega t)$

Exp of periodic time wave → fourier series

Eg converting square wave to sum of sines & cosines. In ppl, take any wave which has a certain period & represent it as a weighted sum of its harmonics

Euler's identity $\boxed{e^{it} = \cos t + i \sin t.}$ $i = \sqrt{-1}$

∴, Plotting a complex sinusoid, cosines are reflection on real axis & sines are reflection along imaginary axis.

anticlockwise ← → clockwise

$$\cos t = \frac{e^{it} + e^{-it}}{2} \qquad \sin t = \frac{e^{it} - e^{-it}}{2i}.$$

$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} C_n e^{\frac{i2\pi nt}{T}}} \rightarrow C_n = \boxed{\frac{1}{T}\int_0^T f(t) e^{\frac{-i2\pi nt}{T}}}$$

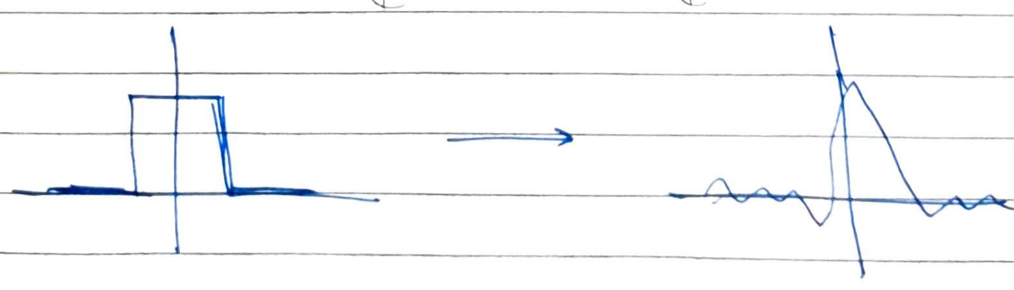
C_n → amplitude / extent to which it's present in signal.

- Fourier Transform : (If f(t) is non-periodic)

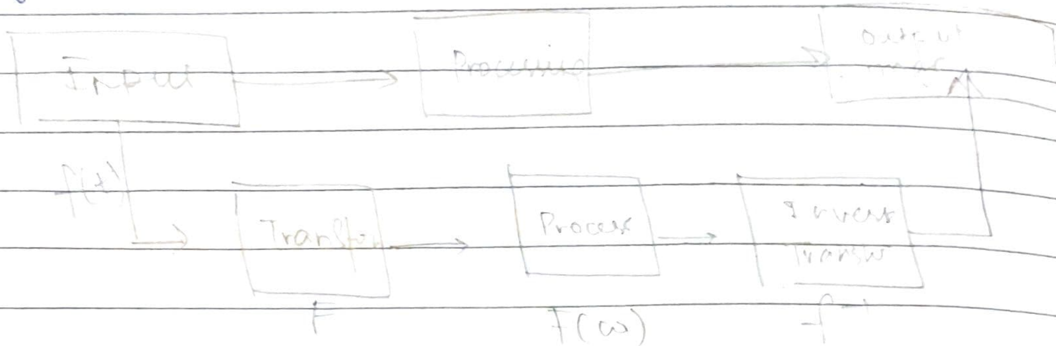
Approx non-periodic signals & approx them with complex sinusoids.

Transform :- $F(w) = \int_{-\infty}^{\infty} \underbrace{f(t)}_{\mathbb{R}} \underbrace{e^{-iwt}}_{\mathbb{C}} dt$ For signal f(t)

$\mathbb{C}$



Magnitude Spectrum : $\sqrt{Re^2 + Im^2}$.

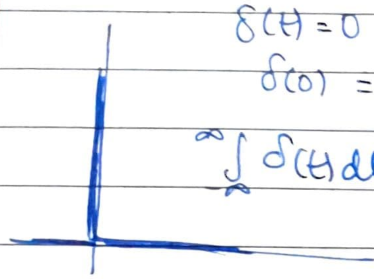


• Impulse function:-

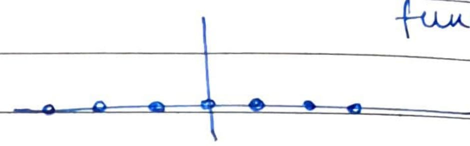
$\delta(t) = 0$ for $t \neq 0$

$\delta(0) = \infty$

$\int_{-\infty}^{\infty} \delta(t)\,dt = 1$



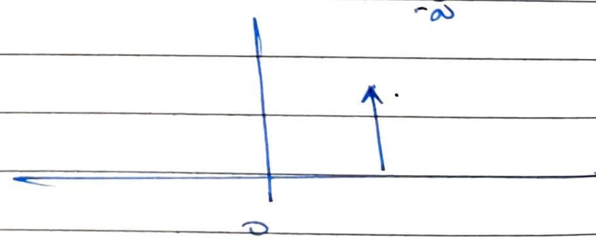
Discrete impulse function



Continuous

↓ can shift by T

$\int_{-\infty}^{\infty} \delta(t-T)\,dt = 1$



→ Properties :-

① $\int_a^b \delta(t)\,dt = \begin{cases} 1 & a<0<b \\ 0 & o/w \end{cases}$

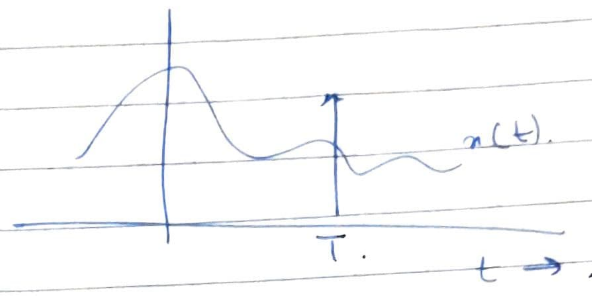
② . $\int_a^b \delta(t) \cdot f(t)\,dt = \begin{cases} f(0). & a<0<b. \\ 0 & o/w \end{cases}$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-i\omega t}\, dt \qquad f(t) = \frac{1}{2\pi} \int_{\omega=-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega$$

Sifting property :-

$$\int_a^b \delta(t-T)\, x(t)\, dt = x(T). \qquad a < T < b.$$

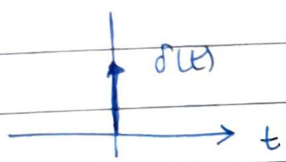
roundabout way.



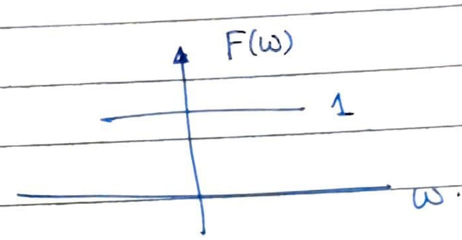
→ FT. of impulse function :

$$F(\omega) = \int \underbrace{f(t)}_{\delta(t)}\, e^{-i\omega t}\, dt.$$

$$\downarrow \quad e^{-i\omega(0)} = 1.$$



⟷



→ Symmetry property :

$$\mathcal{F}(f(t)) = F(\omega)$$

$$\mathcal{F}[F(t)] = 2\pi f(-\omega).$$

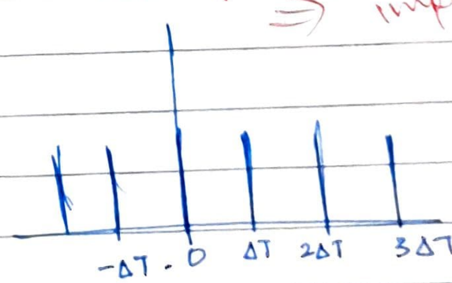
$$\int e^{in\omega_0 t} \times e^{i\omega t}\, dt$$

$$\int e^{-i(\omega - n\omega_0)t}\, dt.$$

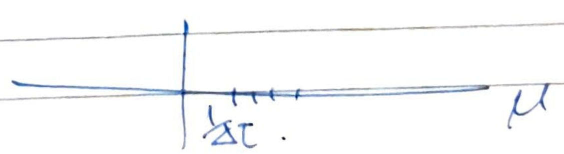
→ FT of impulse train.

Fourier of impulse train ⇒ impulse

$$S_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T).$$



$$\mathcal{F}(\,) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(\mu - \frac{n}{\Delta T}\right).$$



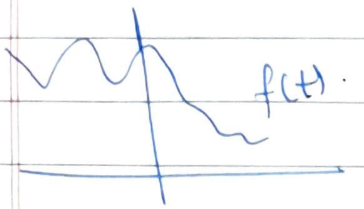
$\mu = 2\pi w_0$

- Sampling = $f(t) \times$ Impulse train.

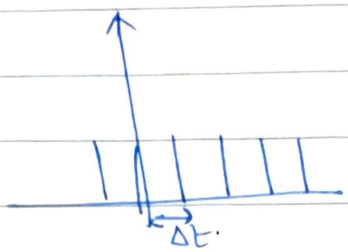
while sampling; the continuous analog signal is multiplied with impulse train

ie ::

$S_{\Delta T}(t) = \sum \delta(t - n\Delta T)$



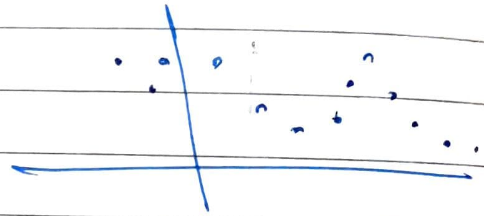
signal.



Impulse Train

convert a set of discrete samples into cont. version

$$\tilde{f}(t) = \sum_{n=-\infty}^{n=\infty} f_n \, \delta(t - n\Delta T)$$



- <u>FT of a sampled function:</u>

$$\tilde{f}(t) = \sum_{n=0}^{n=(M-1)\Delta T} f_n \, \delta(t - n\Delta T)$$

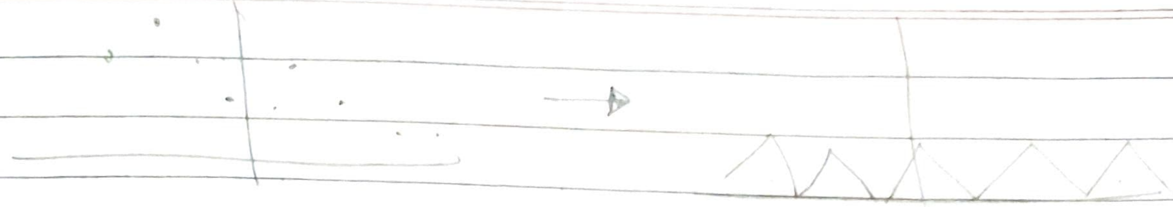
sum of shifted copies of the F

$$\tilde{F}(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(\mu - \frac{n}{\Delta T}\right).$$

$$\tilde{F}(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{n=\infty} F\left(\mu - \frac{n}{\Delta T}\right)$$

- periodic
- continuous

In



$$\tilde{F}(M) = \sum_{n=0}^{n=M-1} f_n e^{-j2\pi M n \Delta T}$$

x ——————— x

11/09/20

Converting non-periodic to complex sinusoids to model real life situation :

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt . \qquad f(t) = \frac{1}{2\pi} \int_{\omega=-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega .$$

→ Sampling : $S_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$.

$f(t) = \left(f_n \times \delta(t - n\Delta T) \right)$ sum.

→ Discrete FT :-

$$F[m] = \sum_{m=0}^{m=M-1} f_n \, e^{-j2\pi n \frac{m}{M}}$$

↳ no direct dep on ΔT

M : length of signal .

$$\frac{m}{M} = \frac{1}{\Delta T} .$$

$F[m] \longrightarrow Re\{F[m]\}$, $Im\{F[m]\}$.

recording energy ...

Phase $= \tan^{-1}\left(\frac{Im}{Re}\right)$ ... strength of the particular angular freq.

Amplitude $= \sqrt{Re^2 + Im^2}$

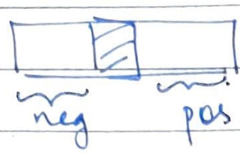
Power F = Power f

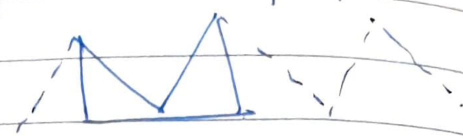
Multiplying $f[x] \times e^{\frac{j2\pi k_0 x}{N}}$ complex sum & obtain shifted DFT result.

Squaring & adding intensities :- power | intensity.

→ Center shifting : move neg & positive freq.

Just for plotting purposes





Can also perform × by comp sinusoid before DFT.

- **Sampling & Freq. intervals :**

Compression in time domain ⇒ Expansion in freq domain.

$f[n] =$ 0, 1, 2, …, $M-1$ (spacing ΔT, ΔT)

∴ total length = $T = (M-1)\Delta T$. … total time period.

$F[m] =$ 0, 1, 2, …, $(M-1)$ (spacing ΔU, ΔU)

→ $\Omega = (M-1)\Delta U = \frac{1}{\Delta T}$ … Range of freq. $\propto \frac{1}{\Delta T}$ → sampling interval

→ $\Delta U = \frac{1}{(M-1)\Delta T} = \frac{1}{T}$ …. freq. resolution of DFT $\propto \frac{1}{T}$

Eg:- Discrete signal : 1 2 4 4 . (at 0, 1, 2, 3, 4.)

$F[m] = \sum_{n=0}^{M-1} f(n) \cdot e^{\frac{-j2\pi mn}{M}}$ $F[0] = 1+2+4+4 = 11$

$F[1] = \sum_{n=0}^{M-1} f(n) \cdot e^{\frac{-j2\pi n}{4}}$

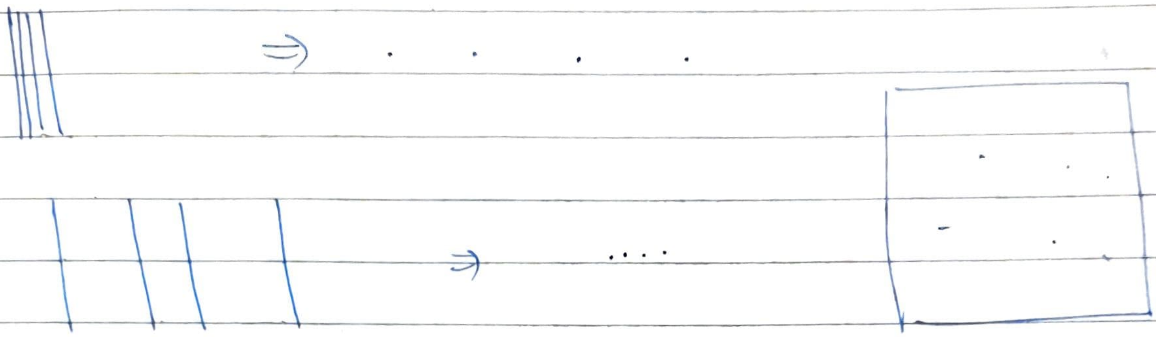
$0 \to 1 \times 1$
$1 \to -j \times 2$
$2 \to -1 \times 4$
$3 \to +j \times 4$
$= -3 + 2j$

Avg freq = . At m=0, n=0, Max value

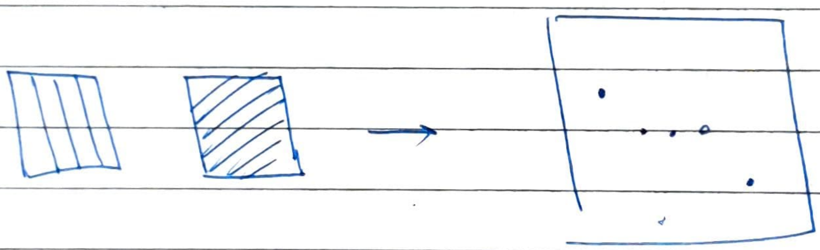
$$\boxed{F[m,n] = \sum_{y=0}^{y=(N-1)} \sum_{x=0}^{x=M-1} f[x,y]\, e^{-j2\pi\left(\frac{mx}{M} + \frac{ny}{N}\right)}}$$

High peak at (0,0) & at wave vector

As freq increases, spatial plots decrease in size.

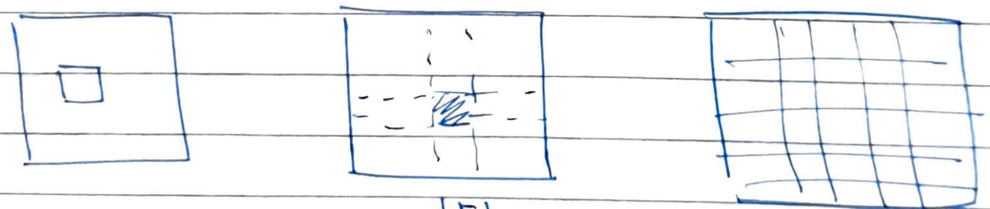


$|F[m,n]|$ → DFT magnitude.



$F(f_1+f_2) = F(f_1) + F(f_2)$.

Eg:-

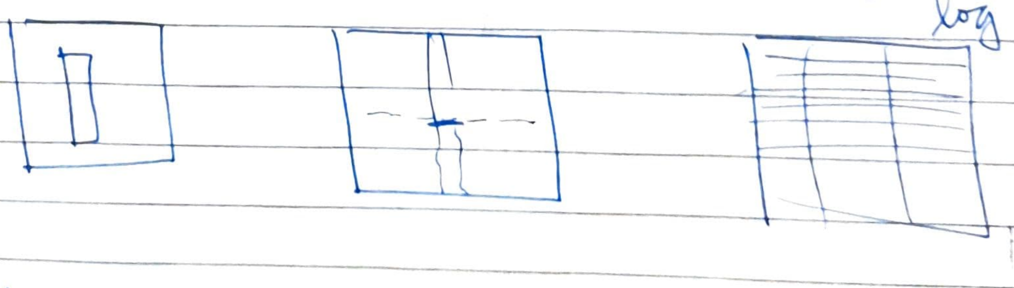


|F|.

Brighter.

log.

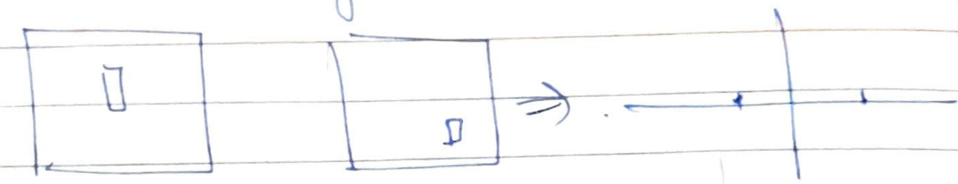
log conversion



M : ht

N : wd.

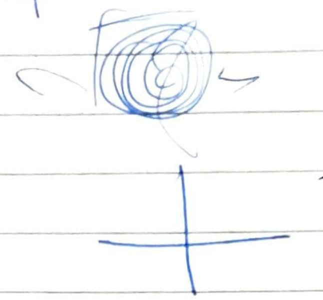
→ Translation & mag remain same.



→ Alphabets. Design patterns

B Q

H T



→ Phase: offset relative to the original

$A\sin(\omega t + \phi)$ ... shift left

→ Phase & spectra both matter for reconstruction.

on performing F | Setting ~~phase~~ spect magnitude ie: $|A(\Omega, \psi)|$ to constant ... reconstruction

inv. F^{-1} | Setting $\phi(\Omega, \psi) = 0$ .... reconstruction not possible

→ phase = 0

|: positional info

N: strength

- Given that we can reconstruct the image with a smaller set of frequencies ⟶ compression

Complexity of DFT $= O(n^2)$

Time red = FFT $= O(N \log N)$.

Ideal Low pass: | Retaining lower change in frequencies ⇒ Smoothing

Apply a mask S.T. only lower freq. retained

Ideal lowpass :- $H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$

As radius inc, smoothing decreases.

1-0 transition causes the ringing effect.
Instead ∴ use gaussian → Gaussian Low Pass Filters.

$H_{hp}(u,v) = 1 - H_{lp}(u,v)$ → Edge detection

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

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

QUIZ :

128 × 128 image   Range $\underline{0-64}$ $64 = 2^6$

6 bits/pixel → 1
128 × 128

6 ×

128 × 128 × 6

1	5	9	6	3	5	6
1	6	6	5	5	5	6

→ Adding 50% of the lowest freq. is not bad & gives us most of the detail.   3 4 5

& High detail ⇒ high freq.

Edges cause high freq.

classmate
Date
Page

2)

4	1	3	2
3	1	1	1
0	1	5	2
1	1	2	2

2	2	2	3
2	2	2	2
0	2	4	3
2	2	3	3

$(L-1) \sum_{j=0} P_r(r_k)$

$(L-1) = 5-1$
$= 4$

	0	1	2	3	4	5
PDF	$\frac{1}{16}$	$\frac{7}{16}$	$\frac{4}{16}$	$\frac{2}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
CDF	$\frac{1}{16}$	$\frac{8}{16}$	$\frac{12}{16}$	$\frac{14}{16}$	$\frac{15}{16}$	$\frac{16}{16}$

4	2	4	3
4	2	2	2
0	2	5	3
2	2	3	3

$0 = 5\left(\frac{1}{16}\right) = \frac{1}{4} = 0$ $\quad \frac{1}{16}$

$1 = 5\left(\frac{8}{16}\right) = 2 = 2 \quad \frac{0}{16} \quad \frac{1}{16}$

$2 = 5 \times \frac{12}{16} = 3 \quad 3 \quad \frac{10}{16} \quad \frac{11}{16}$

$3 = 5 \times \frac{14}{16} \quad 2.15 = 2 \quad 4 \quad \frac{4}{16} \quad = \frac{5}{16}$

$4 = 5 \times \frac{15}{16} \quad 3.15 = 2 \quad 4 \quad \frac{1}{16} \quad \frac{6}{16}$

$5 = 5 \times \frac{16}{16} = 5 \quad 5$

2 pixels

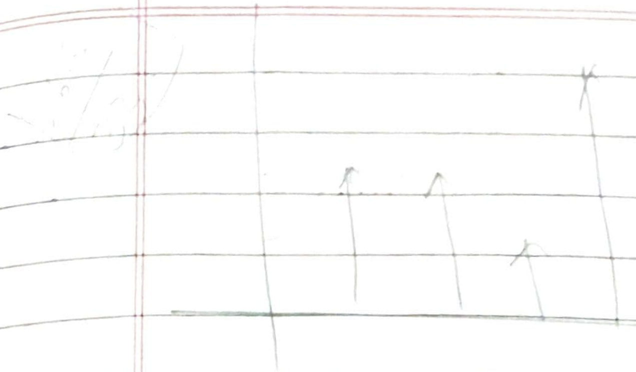
3 3

2 2 2

2x

$$f(x,y) = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{N} F[m,n]\, e^{j2\pi\left[\frac{mx}{M}+\frac{ny}{N}\right]}$$

IDFT



15/09/20. Image Enhancement & Filtering in freq domain.

→ $I \xrightarrow{DFT} F(u,v)\, H(u,v) \xrightarrow{IDFT} I_{LPF}$.

$\times$ in the transform domain $\equiv$

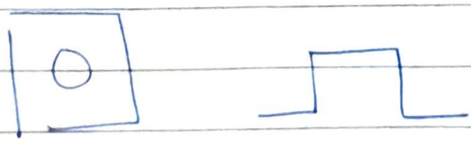
ie: one can convert $H(x,y) \xrightarrow{IDFT} h(x,y)$
& convolution on the $I * h(x,y)$.

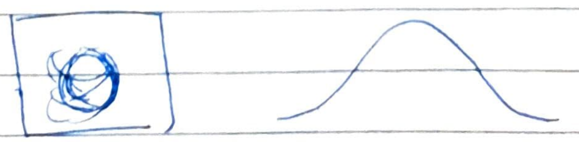
Other option: butterworth LPF

→ The radius is the dial that is used to increase power
ie: amount of total freq. content present in the image.

Jump in freq. is not that much.
10, 30, 60, 160, 460.
87, 93.1, 95.2 . . . 99.2

→ Ringing effect is due to the higher level harmonics

For eg:  Stark difference.

 Gaussian low pass filter instead!

Gaussian smoother too much
⇒ Butterworth achieves middle of ground

Additional higher freq. components do not exist. Ringing effect dec.

Application of low pass filtering
- Text breaks can be fixed with the help of LF.

Low pass: define a mask such that only those frequencies that you want are retained.

→ High pass :- $H_{hp}(u,v) = 1 - H_{lp}(u,v)$

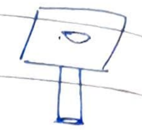
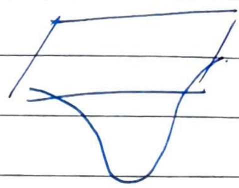


Image sharpening

Edge inf inc and is more clearly visible.
As $D_0 \uparrow$, edge info increases, this is because only higher frequencies are retained ⇒ high freq are retained.

Again, we have gaussian Hp :-



- Laplacian in Freq. Dom:

$$\mathcal{F}\left(\frac{d^n f(x)}{dx^n}\right) = (ju)^n F(u).$$

$$\mathcal{F}\left(\frac{\partial^2 (f(x,y))}{\partial x^2} + \frac{\partial^2 (f(x,y))}{\partial y^2}\right) = (ju)^2 F(u,v) + (jv)^2 F(u,v)$$

$$= -(u^2+v^2) F(u,v).$$

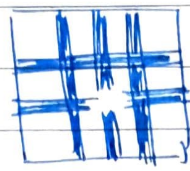
→ Notch Reject Filter (NPass Filter)

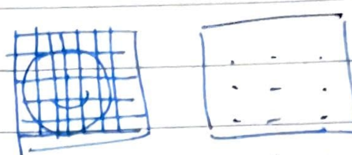
Characteristic of retaining central freq.

Specify a custom mask.

→ Artefact removal.

Create a mask :





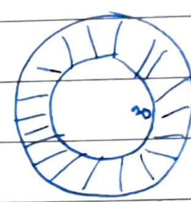
grid pattern
freq. domain

Unsharp : $I - I_{LPF}$.

High boost : $I + c\nabla^2 I$

→ Band Reject Filters :

After sufficient observation we see that we do not want the freq. values $20 < x < 40$.



→ ~~Fuiss~~ Homomorphic Filtering :

$$I(x,y) = \underbrace{M(x,y)}_{\text{material}}\ \underbrace{L(x,y)}_{\text{light}}.$$

: depends on intrinsic prop. of material & light.

$$\log I(x,y) = \log(M(x,y)) + \log(L(x,y)).$$

Perform freq. domain processing individually.

→ Performing circular conv. → wrap around error.

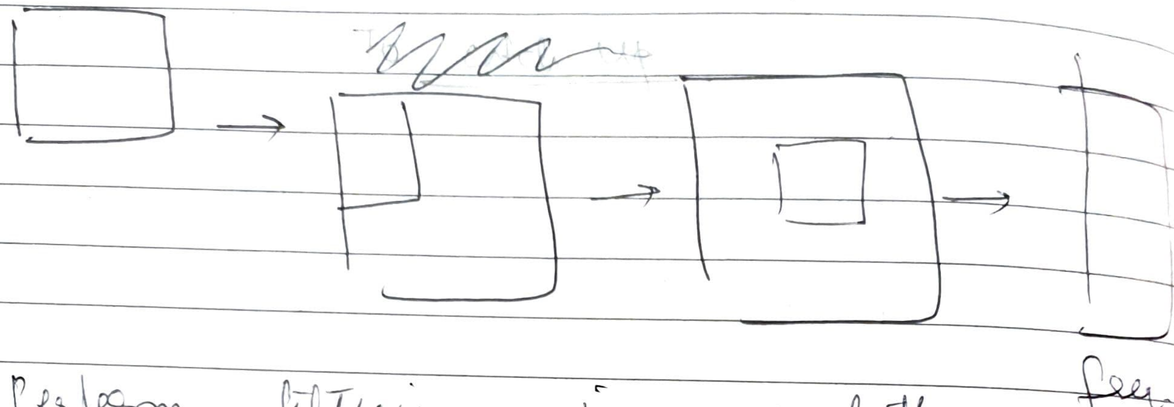
Simple recipe :- zero padding.

$I \xrightarrow{DFT} F$

FH.

Steps: Given $M \times N$ image f.

1) Pad f_p to size P, Q $P = 2M$, $Q = 2N$.
2) Mult f_p by $(-1)^{x+y}$ ... centering.
3. $F_p = DFT(\tilde{f_p})$



Perform filtering, inv, reshift:

Algo:-

read . im

sobel = 3 x 3

imf = fft (im, 402, 402) .

Sf = fft (sobel , 402, 402)

F_fH = shift(imf) * fftshift (sobel)

ifft (FH .

→ Only linear spatial filters can be transformed to frequency domain for processing.

→ Not non-linear (min, max, median)

Guide process of spatial filter design.

Gabor filters, wavelets, shape descriptors.

- **Morphological Processing:** (study of shape).

— Binary images : thresholding : one kind of contrast stretching.

Used in plant phenotyping.
→ shape of leaves
→ count number/kinds of leaves.

— Background subtraction.

- **Morph. Operators:**

While performing : assu: ∃ a $\underbrace{\text{region of interest}}_{\text{set of pixels}}$.
0 : background
1 : foreground.

interest : fg. everything els : bg.

- Structuring el :- → Binary numbers

Box.

1	1	1
1	1	1
1	1	1

Specify se = strel($\underbrace{3,3}_{\text{size}}$, disc) → shape.

Dis

0	1	0
1	1	1
0	1	0

SE
- size
- shape
- origin

Pencil & eraser tips are examples of this

Origin : default = centre. (can be top left).
1 : active els
0 : don't care conditions.

First Type: Erosion

Thinning kind of op for binary structure

22/09/20.

Bin Images :- Objects/ Regions
→ 0 bg 1 fg.

Thickening or erasing
Shape of the tip : determined by structuring el.

Effect of struct el. in erosion is ∥ to filtering.

Algo: If ∃ location ST: all the locations completely get covered by S.E then retain, else erode.

→ Reduces no: of foreground pixels.

→ Output of erosion = $\boxed{f(\text{s.el used}) \text{ \& origin}}$ of the s.el.

Erosion shrinks actual obj but tends to enlarge ~~the~~ holes within the object.

Counting coins eg: Binarize (threshold) → Erode the edges

As disk size increases, erosion increases

→ Erosion can be considered similar to (Min filter)

→ I3 = imerode (I2, SE). → code.

dil b/w spatial filters - average binary

→ Simple application of pattern matching:
Find all locations where ∃ a collection of pixels that matches SE.

- ie: Erosion recap:-
  - Shrinks foreground
  - Foreground holes enlarged
  - Small obj / noise increase

Useful to remove noise.

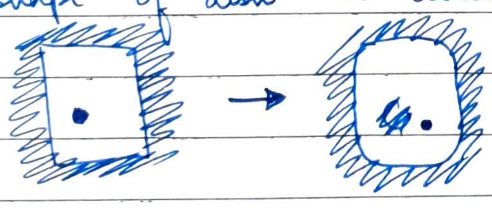
- $\boxed{\text{Dilation}}$:

Opposite of ~~st~~ erosion.

Algo: If ∃ at least 1 pixel that touches the filter/SE, set all to FG.
Thickening operation

Holes shrink, Foreground thickens.

SE shape of disk is used ⇒ rounding of corners occurs.



Diameter of circle is smaller for the black dot

a) Disk SE b) Multiple dilations applied.

Dilation can be considered as a (Max Filter)

→ ~~Lea~~ Text breaks can also be fixed via this

- Expands fg.
  fg holes shrunk.

text breaks fix — LPF, Dilation

→ Applications of erosion & dialation :- boundary detection

Dilation

Steps :-

1. Dialate input
2. Dilate − Input

↳ Result is only boundary

Can use erosion also to do so.

1	1	1
1	1	1
1	1	1

→ 8-connectivity

All 8 pixels have some kind of connectivity

8 : thicker

4 : thinner, diagonal connect not considered here.

0	1	0
1	1	1
0	1	0

All operations now are just combinations of the 2 fundamental operations

- **Opening**

Same struc. el.

Slide the SE inside the foreground, remove pixels that don't fall

Can be considered as <u>Erosion followed by dilation</u>

Helps preserve the shape of the object except for the narrow parts

→ remove isolated islands
→ smoother portions.

App: used in filtering out certain shapes.

(!) Opening is idempotent : repeated application has no effect

Opening = Erosion → Dilation with same structural element.

• Closing : → thin regions get closed

Do dilation followed by erosion.

Application: Skeletonization looks better for closed images.
backbone is much better for closed images

Here, slide the structural element outside this line

Opening ⇒ dual ⇒ closing. (I) narrow inlets get filled

Opening foreground ⇔ closing bg (1-I)

eg → Fingerprint scanner
Opening + closing] remove all tiny patterns.

Dilation & closing : adding fg pixels
Erosion & opening : removing.

~~⊖ : dilation~~ ⊖ erosion
∘ opening
• closing
⊕ Dilation

narrowing extending

~~$(f \oplus s)(x) \leq f \oplus s$~~

counts → $(f \ominus s)(x) \leq (f \circ s)(x) \leq f(x) \leq (f \bullet s)(x) \leq (f \oplus s)\, x$.

erosion opening closing dealition

sets → $F(f \ominus s) \subseteq F(f \circ s) \subseteq F(f) \subseteq F(f \bullet s) \subseteq F(f \oplus s)$

- Erosion on gray value images
  → min filter → eroding a gray image
  → images tend to get darker. → pulls to darker

- Dilation on gray
  - Max filter → pulled to lighter
  - More uniform intensity
    → small isolated intensities getting magnified

25/09/20

HAM

- HIT & ~~TRIAL~~ MISS TRANSFORM:

→ Eg: finding right angle convex corners

→ St. els:

x	1	x
0	1	1
0	0	x

→ don't cares

Rotate to get all corners & convert to A||B||C||D.

Slide st. el across the entire binary image.

Eg:-

	1	

a) Isolated points.

0	1	0
0	0	0

b) end points on thin lines

	1	
	1	
1		1

1		
	1	
1		1

*	0	1
1	1	0
*	1	*

Locate triple junctions

- Distance transform.

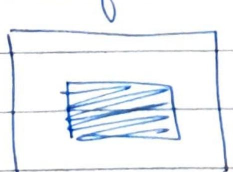
for bin images:- compute the dist to bg.

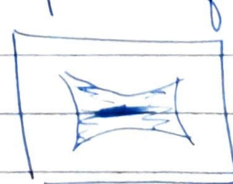
Chessboard dist :- $p_1: (x_1, y_1)$ $p_2 (x_2, y_2)$.

$$D(p_1, p_2) = \min \{ |x_1 - x_2|, |y_1 - y_2| \}.$$

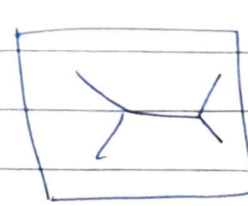
This dist transform is made use of in skeletonization.

Can be defined as points of local maxima





Dist transform



skeleton

Skeleton :- points of local maxima within the image.

Date
Page

Types of skeletonization:

TYPES

- by thinning : not reversible.
- chamfer (3,4) no pruning : fully reversible
- Chamfer (3,4) with pruning (not full rev)

- Much more compact representation. (signature)
- properties of the shape
- Can be used in pattern matching.

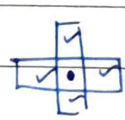
- Finding Components:

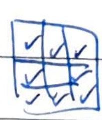
Quick motivation

Consider the different touch screen devices, want to detect no: of fingers touched
· Or gesture recognition

Plant phenotyping → shape & no: of leaves.
Granulometry → pack ~~diff~~ grains of same size

Connectedness :-

4-connected :-  only top bottom RL.

8-connected,  All 8 squares

→ 2 pass algo for conn. comp. labelling.

White as the object

For BG

If left pixel is entered already, cp label to next pixel
If left pixel is bg, give new label #

complexity :- $O(MN)$ + disjoint DFS: $O(V+E)$

For next row, cp top labels.

If left pixel & top pixel are same, choose smaller ID & make the larger, the child of smaller.

- 1st pass generates im with multiple IDs.
- 2nd pass : replace child label with root label.

- Flood fill:

similarity → DFS paint

Bucket fill tool in MS paint. Fill pixels based on the given pixels & boundaries

Specify a seed component :- recursive algorithms.

Based on connectedness the flood fill occurs

- GEOMETRIC OPS:

→ Manipulates the ht & wd of the image | shape. Change img geometry.

Eg: translation, rotation, scaling etc.

Procedure for translation of images now :-.

$x \rightarrow f_x(x, y) = x'$

$y \rightarrow f_y(x, y) = y'$.

And $I(x, y) = I'(f_x(x, y), f_y(x, y))$.

Combination

Eg:- Scale, Rotate, Flip, Translate, Affine

- Translate:-

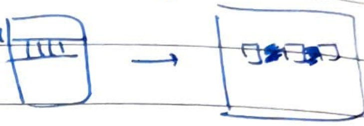
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}.$$

- Scaling:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Stretching creates holes in the image

contraction : collision amongst intensities



Sol:-
- subsampling : single pixel selection
- interpolation : (avg. of neighbouring image)
- replication - copy
- interpolation -

- Shearing:-

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & bx \\ by & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

add small value of y to x.

- Rotation :-

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

- Image warping :-

$$\boxed{x' = x + y * \sin(\pi * x).}$$

- Homogenous coods.

$\rightarrow$ point x matrix

converting all ops. to $\boxed{\text{point - matrix}}$ mult.
Translation problem with mat mult.

$\therefore$ Add a dummy Cood.

Combinations can be done with single matrix

Add a dummy coord here

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \longrightarrow \text{gets desired result.}$$

Homograph
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & & \\ h_{31} & \cdots & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$\rightarrow$ Affine combination of scaling, rotate, transform.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

$\hookrightarrow$ fixed

Inverse transform is inverse mapping

Affine $\Rightarrow$ includes all

29/09/20

$\rightarrow$ Filters, Point Op:
- $\rightarrow$ Modify colour vals.
- $\rightarrow$ Domain (mostly fixed).

$\rightarrow$ Geo op:
- modify position of pixels.
- Colours kept (mostly) same

Defining functions modifying co-ordinates.

ie: $I \begin{cases} x \rightarrow f_x(x,y) = x' \\ y \rightarrow f_y(x,y) = y' \end{cases} I'$

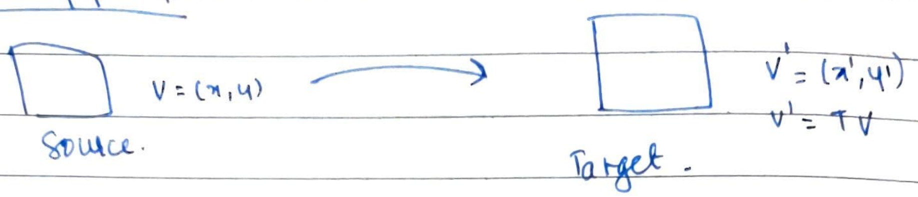
- Affine : combination : $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

easier to perform operations.

Inverse of transform matrix is ∴ inverse mapping.
And hence ⇒ affine mapping is invertible

- Interpolation Methods:

→ Forward mapping :- / Forward warping



Issue :- Non integer v' values cause issues.
Co-ordinates may go beyond image bounds.

Dimensionality & - Discretization issues.

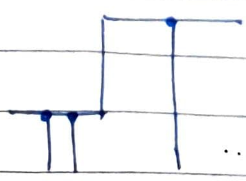
If we want to retain the entire image content, then we have to determine the size beforehand.

→ To overcome this we use
Backward mapping:

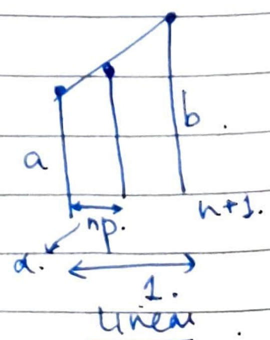
Go from the target image to source.
Sample value obtained via interpolation [∵ original floating
interpolate

→ Interpolation functions :-

$$I_p = (1-\alpha_p)\, a + \alpha_p b$$



∴ 1D nearest neighbours.

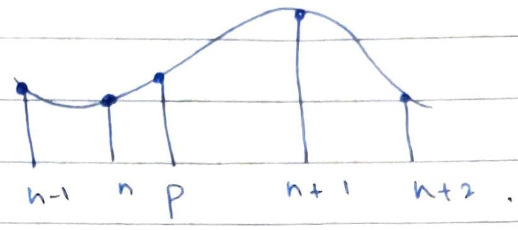


~~10/~~ NN → preserves colours

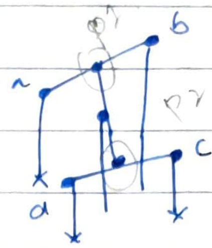
classmate
Date
Page

colours retained better

Cubic Interpolation
Intensities vary as a cubic function of the neighbours.



Bilinear :-
2 α , 2 β values



: new colours 'can' be introduced.

NN is quite step-y :- hence ∃ some artifacts.

Eg: imrotate (Im, 30°, loose/crop) anticlockwise.
loose → change shape; crop → retain shape.
anticlockwise ↓ ~~retain~~

G. Transform
- Mapping Type of transform.
- Interpolation

→ Cross dissolve :
Morphing one persons image onto another.

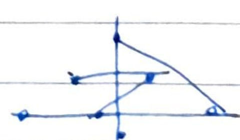
$I(t) = (1-t)*S + t*T$ source $t=0$, target $t=1$ } param.

0 t

1. t: no: of steps. For each step we get an intermediate image.
Issue : ghosting effect.

Mesh based :- at every stage : valid image based on anchor points

$x_s' \ y_s' \rightarrow x_T' \ y_T'$: use cross dissolve.



Anchor points to be selected for both source & target.

for a general pixel point look at dist from all anchor points wrt

Image Reg. Morphing:- Given O & I, find T.

Again find anchor points

→ Geo Op Uses
- correct distortions introduced during imaging
- Transformations: special effects.
- Registration: register 2 images taken of the same scene at diff. times

Intensity $= \frac{(R+G+B)}{3}$

COLOR IMAGE PROCESSING:

VIBGYOR band] White light on a prism.
ultraviolet. → infrared] 400nm – 700nm. visible.

→ Phy. Props:- colour depends on: source, surface, reflect

Radiance (W), Luminance candela/m², Brightness : intensit

– Cones responsible for colour vision. → photopic. LMS RGB
– Perception of a colour is a combination of illuminati & reflectance. $I(x,y) = Il(x,y) * Ref(x,y)$.

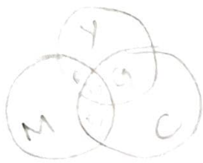
$0.299R + 0.587G + 0.144B$

– Intensity is a weighted function of rgb. & the perception depends on these weights
– Perception of colour is nervous system consequences, not inherent in wavelengths.

nothing as colour

Diff people diff. levels of colour receptors. Tetrachromat
Mantis Shrimp is one species.

$$\boxed{\text{Perc. colour} = f(\text{light, object})}$$



Perceived colour

$= f(\text{Light}, \text{Obj})$.

Eg: Red Objects in blue light is appears black.
Brightes colours in colours channels.
Eg: Red strawberries brightest.

Paints x printing use: subtractive colour scheme

X ——→ X

→ Radiance : total energy flow from light source. (W)
Luminance : amt of energy an obs. perceives candela/m^2.
Brightness : intensity : sub. measure.

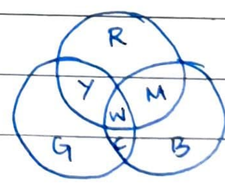
- colour as a physiopshycological phenomenon.

- what we perceive as colours depends on illumination, reflecting wavelengths. ie: $I(x,y) = Il(x,y) * Refl(x,y)$.
- Cone sensitivity : diff a diff extens.

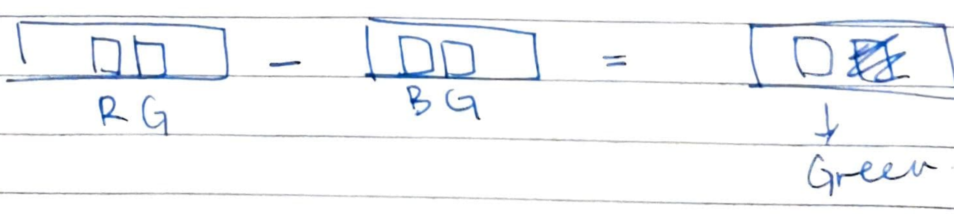
ie: Stimulus $\xrightarrow[\text{cones.}]{\text{processed by}}$ Product | Result

Perception : $0.299R + 0.587G + 0.114B$ wts.

- Primary colours:
  - Additive (CRT displays).
    offer good contrast to dark screens.
    combines on a surface that reflects all light falling on it.



- Subtractive filtering: CMY → primary cols.



$M = R + B$

$C = B + G$

$Y = G + R$

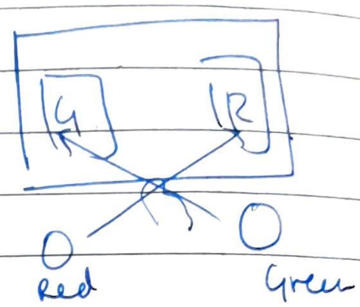
→ CMYK in printers cause inks don't emit, they only absorb. Cannot get true colours easily.

Cyan = $W - R$ Magen = $W - G$ Yellow : $W - B$.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

K : Neg. of intensity image

→ Projector systems



- Colour Models

① RGB :- $f(x,y) = \alpha_1 R + \alpha_2 G + \alpha_3 B$. $\alpha_i \in [0,1]$ $\Sigma \alpha_i = 1$

Perc. not uniform. Cannot differentiate b/w greens.

② Chromaticity diagram CIE :- Rim of CIE diagram represents pure

separate the luminance + chrominance. Yxy ← rel. lum. ; xy chrom. coods

Y : chosen s.t. matches lum eff. func. of human eye.

- McAdams ellipses :- cannot diff b/w colours in ellipse & relative to center.

Define LAB colour space from chromaticity diag & Mc Adams ellipses (L^*, a^*, b^*).

Color dis LAB $(C_1, C_2) = \|C_1 - C_2\|$.

$$= \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$$

Poor choice of colour cmap dooms what you want to convey ⚠

Whitening $Y = AX$.

Chose $A^T A = \Sigma^{-1}$ .... cov. matrix.

- HSI : hue, sat, intensity

H: dominant col. perceived by person → angle

S: White light mixed with hue. pure : saturated (1)

I: Notion of intensity. [0,1] → 255

Saturation : Relative purity.

HSI : color description   RGB : generation.

09/10/20.

→ CRT displays : additive.

↳ contrast on dark images.

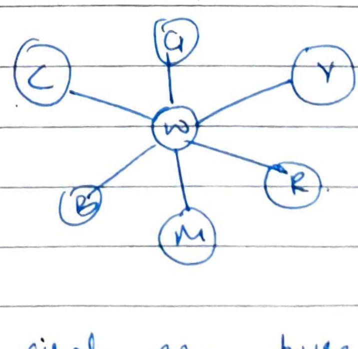
→ CIE lab space changes in CIE space lin. wrt human perception.

→ Color Gamuts : - colour printers, colour monitor.

- Choose colour gamut correctly for faithful rep

- area : range of colours

→ Prim + Sec :-



: Pure colours

Hue circles.

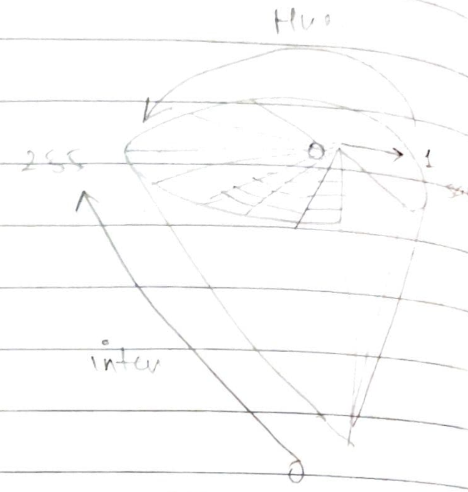
- At the rim of the circle are pure hues (sat = 1).
- Hue dominates less as move to center.
- center no saturation

RGB to HSV :-

$R' = R/255$ $\qquad$ $C_{max} = \max(R', G', B')$

$G' = G/255$ $\qquad$ $C_{min} = \min(R', G', B')$

$B' = B/255$ $\qquad$ $\Delta = C_{max} - C_{min}$.



$$H = \begin{cases} 0 & \Delta = 0. \\ & \end{cases}$$

$$S = \begin{cases} 0 & C_{max} = 0. \\ \dfrac{\Delta}{C_{max}} & C_{max} \neq 0. \end{cases}$$

$V = C_{max}$. ie : HSV & HSI diff is :-

for $I = \dfrac{R' + G' + B'}{3}$.

extrapolate the triangular crossection in the GUI.

→ White balancing : ambient lighting.

→ Pseudo colour :- use floating point values thereby increasing the space significantly

- Items close to camera :- white, far : darker

  Depth image : rounding off uniform bg.

- Colour transforms used in security check. Diff objs. show up differently.

f(x,y) → R Trans → $f_R(x,y)$

f(x,y) → G Trans. → $f_g(x,y)$

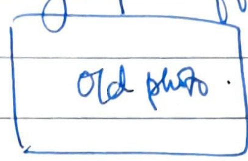
f(x,y) → Blue Tran → $f_b(x,y)$

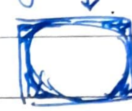
Sinusoidal transformations. Look at diff combinations of R, G, B to detect explosives, map to diff things.

→ Also used for satellite imagery. Multispectral im proc.

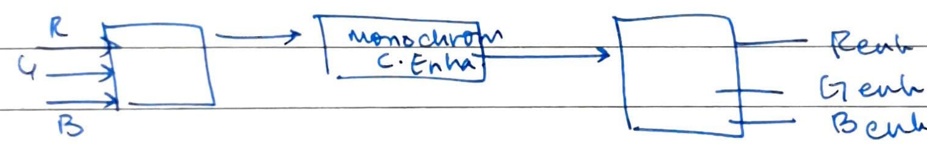
Other imp aspect while dealing with RGB is A: transparency. $\alpha \downarrow$ trans. $\uparrow$. Alpha channel.

→ Applications: Filters, Vintaging effect, Vignetting effect.

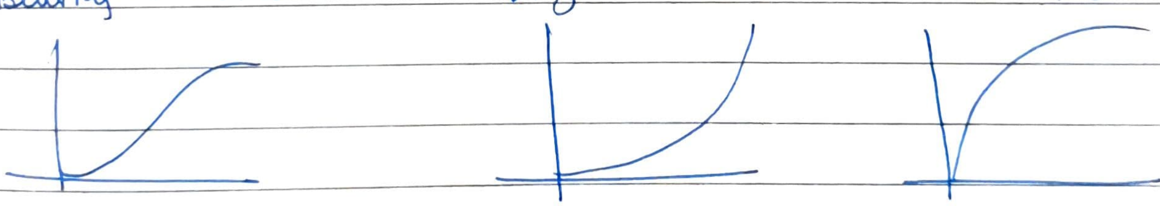




→ Contrast enhancement.



→ Blurry → normal. Bright → Normal Dark. Normal



• Hist Eq. Colour:

Opt 1: Equalize each channel independently. And the interpolate.

Opt 2: Take Avg. hist of R, G, B - Equalize WRT. Avg hist.

Opt 3: Ref Image = I of HST. (may not correspond to perception of intensity)

My opinion :- Avg. Opt2 'best'.

- Grayscale quantization :
  Fewer colours — quantization .

Color quantization on RGB :- K-means ~~of~~ on RGB .
Center of this vol. is the reference colour .

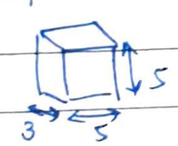
→ Median filtering . now its not purely pepper noise.
- Scalar adaptive filtering .
- RGB or HSV ... what to choose .

- <u>Colour Edge Detection</u> :

App 1: Image → R → Grad R — Edge R → EdgeMap .
Image → G → Grad G — Edge G → EdgeMap
Image → B → Grad B — Edge B → EdgeMap

Locations in image where ∃ aprupt change in clr.

CNN : first set of filters learnt : low level features.
Filter



13/10/20

- <u>IMAGE SEG</u> :

Partitioning an image into a collection of connected set of pixels .

a) Into regions cover the image
b) Structures (lin) → line curve.
c) 2D shapes — circles, ellipses, ribbons.

Sobel : smoothen / avg the image followed by edge filtering [i.e., avoid noise]

$\frac{1}{4}[1\ 2\ 1]\begin{bmatrix}1\\0\\-1\end{bmatrix}$

13/10/20

Edge based

3 basic grey level discontinuities : detection of discont.

- Points / corners
- edges
- lines

} abrupt change in intensity

result has edges as well

→ To detect a single point whose int. differes from neighbours. Fires maximally at point based regions. ~~LED~~ LED blinkers. Blinking, LED detectors. threshold output

2D derivative

→ Image gradients & Edges : in X direc & Y direc

1st derivative : edge detection   2nd deri : edge direction

Edge ⊥ to grad

Edge detection : prewitt, Sobel, laplacian etc.

Combined edge image.   2D filters accumulate more evidence abt presence of edge.

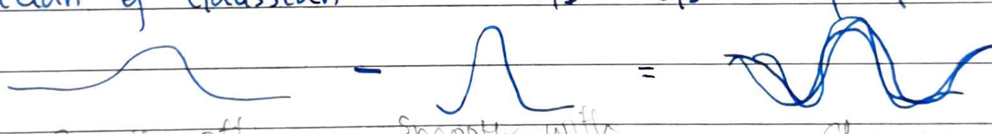
→ $|g_x| + |g_y|$

or → $\sqrt{g_x^2 + g_y^2}$.

$\boxed{\theta : \text{orientation}\ \ \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)}$

→ Laplacian of Gaussian :- $G_1 - G_2$ ⇒ profile as edges

some eye features behave like this



smooth with 1st gaussian   smooth with second gaussian   Cheaper version of laplacian

- Oriented Line detection : can use sobel X, Y to obtain horizontal & vertical edges.

→ For +45°.

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

H.T To detect more detailed dimensions - lines, circle, alg curves etc. :- Hough Transform.

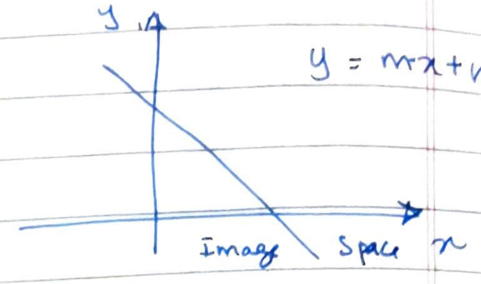
can use $g_x \begin{bmatrix}-1 & -1 & -1\\ 2 & 2 & 2\\ -1 & -1 & -1\end{bmatrix}$ horizontal $g_y \begin{bmatrix}-1 & 2 & -1\\ -1 & 2 & -1\\ -1 & 2 & -1\end{bmatrix}$ directly

LECTURE 17:

- Hough Transform:
  Use for ~~lines~~, curves, algebraic curves, arbitrary but specific shapes.

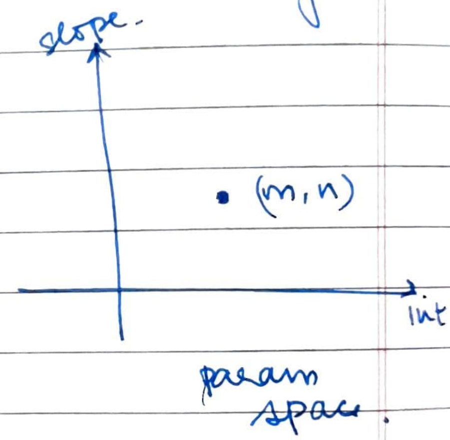
Relies on notion of parameter spaces.

Consider we have an image performed edge detection & thresholding.



∴, Pixels corresponding to edges exist in this image.
Line in image can be parameterized by
$y = mx + n$

At least 2 points or slope & intercept characterises the line.

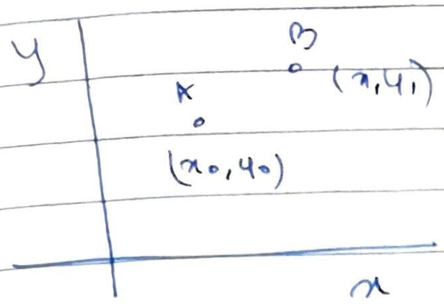


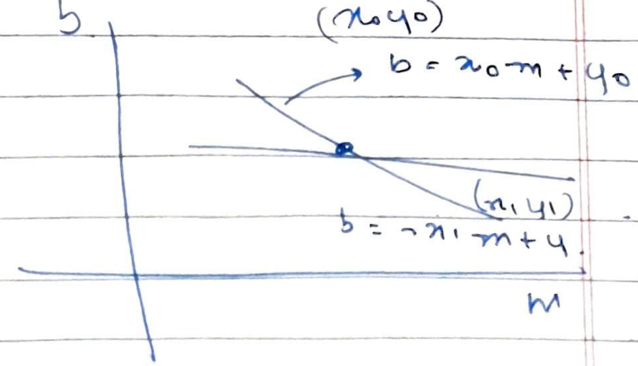
Line $\overset{\text{dual}}{\Longleftrightarrow}$ Point.
Points $\Longleftrightarrow$ Lines (many).
Co-linear pts $\Longleftrightarrow$ Intersecting lines.
(point of intersection is the line).

→ HT: Given an edge point $\exists\ \infty$ no: of lines passing through it (vary m, n)

Each of these lines can be represented as point in param space

If $\exists$ 2 points (x_0, y_0) & (x_1, y_1).

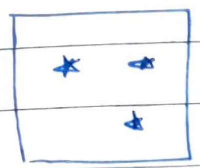




The lines intersect at the params given by lines joining A & B. in the Hough Param Space.

Eg:

I.

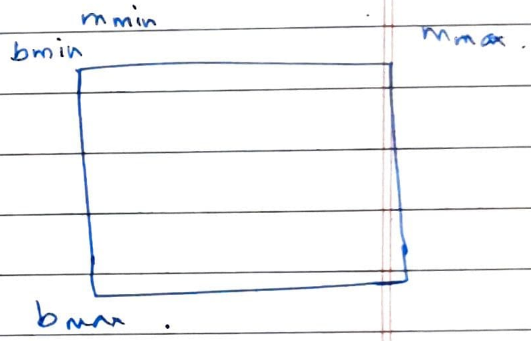


to find all possible lines : v. high complexity.

Image space

→ Linear algo alternative (HT).

Initialize an accumulator array $A(m, b)$ to 0.

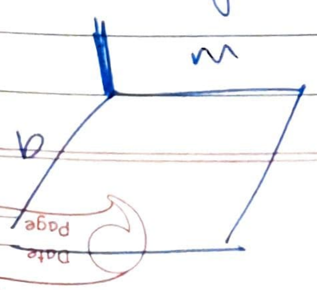


For each el. increment ~~vat~~ locs. that satisfy the eqn ~~to~~ $b = -xm + y$ in this mapping.

∴ All pts for an edge will inc. a particular m, b location.

After doing this the local maxima correspond to lines in edge image.

Once we know these vals, we can draw our lines by setting top left as origin



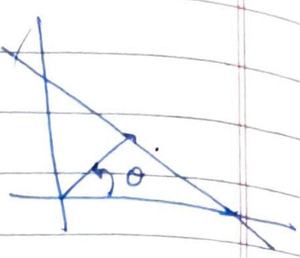
single line of peak corresponds to the line detected

Issues :-

→ slope of line is $-\infty < m < \infty$.
Param space is infinite.

→ Rep of $x = k$ doesn't exist.

Sol:- use <u>normal</u> eqn. of line



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

Now $\boxed{\rho \ \& \ \theta}$ are our params equivalent to m & n.

⇒ Point in img space becomes $\boxed{\text{sinusoid}}$.

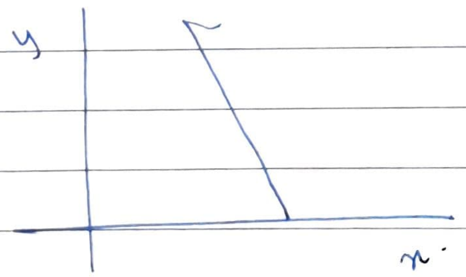
$0 < \rho < D$
$-\pi < \theta < \pi$.
} FINITE SPACE

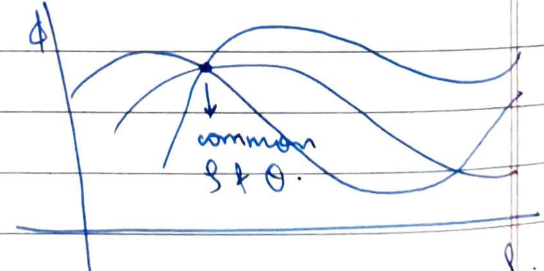
∴ new space can represent all lines

<u>Algo:</u> Edge image $E(i,j) = 1$ for edges).

1. discretize θ & ρ in increments of $d\theta$ & $d\rho$ to get the accumulator array. ρ 0 ... D
2. For each pixel $E(i,j) = 1$ $h = 1, 2, \ldots T$
    1. $\rho = j\cos(h * d\theta) + i\sin(h\,d\theta)$.
    2. Find closest int k closest to this ρ.
    Increment $A(h, k)$.
3. Find local maxima now!

eg:





Dealing with proble

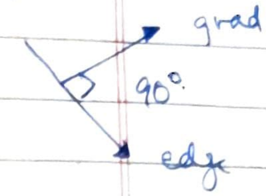
HT speed up:-

→ Edge image < loc, mag, orientation.

∴, Orientation of grad $\tan^{-1}\left(\frac{G_y}{G_x}\right)$.

∴, uncertainty about θ is gone.

∴, fix θ in param space



Generalizations:

→ locate circles etc.

Circles: $(x_1-a)^2 + (x_2-b)^2 = r^2$ 3 params a, b, r.

3D accumulator array.

& for each point on circle edge (x, y) increment $A[a, b, r]$:- point on circle vote for that particular combination.

↳ param $a = x + r\cos\theta$ $b = y + r\sin\theta$.

Optimization: Only 1/8 of circle needs to participate.

Conc: As long as $\exists$ a parametric form for a shape, we can use Hough Transform [As dim increases accumulator complexity increases].

Problems: - It's blind

- susceptable to noise.

→ ★ Map a difficult pattern to a simple peak detection problem.

→ Segmenting out lines :- corresponding to lines & not lines.

- Works through occlusion.
- Any part of st. line can be mapped.

Harris Corner Detection :

HAM prone too noise.
Visual system uses discontinuities as proof

Idea: Easily recognize point by looking through small window

∃ must exist large change.
- → Move in constant intensity : no change (anywhere)
- → Edge: along edge no change in intensity. ↕
- → Corner: significant change in movement. (Intersection of edges)

Math:
$$\boxed{E(u,v) = \sum_{x,y} w(x,y) \left[I(x+u, y+v) - I(x,y) \right]^2}$$

For the shift $[u,v]$ — window func. — shifted intensity — squared change in intensity

→ will apply to above 3 cases & see results

Now, for small shifts, $[u,v]$ we have bilinear exp.

$$I(x+u, y+v) \approx I(x,y) + (uI_x + vI_y) + \frac{1}{2}(I_x^2 u^2 + I_y^2 v^2 + 2 I_{xy} uv)$$

$f(x+h) = f(x) + hf'(x) + \frac{1}{2} h^2 f''(x)$

$$\therefore, \quad E(u,v) = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}.$$

where M is a matrix computed from image derivatives

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}.$$

Take M & use to classify image points using eigen values of M. λ_1, λ_2.

★ λ_1, λ_2 are small in flat region :→ since I_x^2, I_y^2 grad in x, y direction won't change in x & y direction.
E almost constant in all directions.

★ Edge: Either $\lambda_2 >> \lambda_1$ or $\lambda_1 >> \lambda_2$.
Either x & Y grad.

★ Corner: λ_2 & λ_2 large $\lambda_1 \sim \lambda_2$.
E inc in all directions.

Now, to come up with a cornerness response :-

$$\boxed{R = \underset{\lambda_1 \lambda_2}{\det M} - k(\underset{\lambda_1 + \lambda_2}{\text{trace } M})^2}$$

If R is vv large → corner.
If |R| vv small → flat.
If R large negative → corners

→ Find pts with large conere response & take pts that are local maxima.

Use Being able to warp - map images is useful.
→ Compute cornerness response
→ Perform thresholding
→ Take pts of local maxima

LECTURE 18

Image segmentation : partitioning image into collection of connected set of pixels
→ Regions , linear structures etc.

→ When to say an image is partitioned
Partitioning image I into set of regions S satisfying

1. $\bigcup S_i = S$ .... partions cover whole image
2. $S_i \cap S_j = \phi$ ... No regions uliessed .
3. $\forall S_i \; P(S_i) = \text{True}$ .... Homogeneity predicate (critiria $\exists$).
4. $P(S_i \cup S_j) = \text{False}$ .... Union of adjacent regions

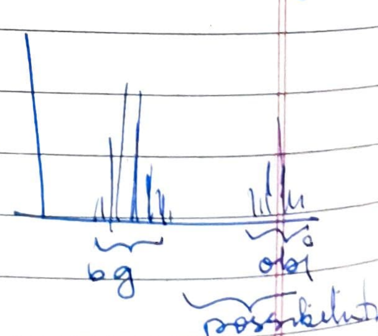
Used to define a valid segmentation.

→ 2-Class segmentation :
~~Tw~~ Convert a grayscale to binary image.

easy way : thresholding : $g(x,y) = \begin{cases} 1 & f(x,y) > T \\ 0 & f(x,y) \leq T. \end{cases}$
T const over entire image : global
T changes variable thresh. /adaptive thresh

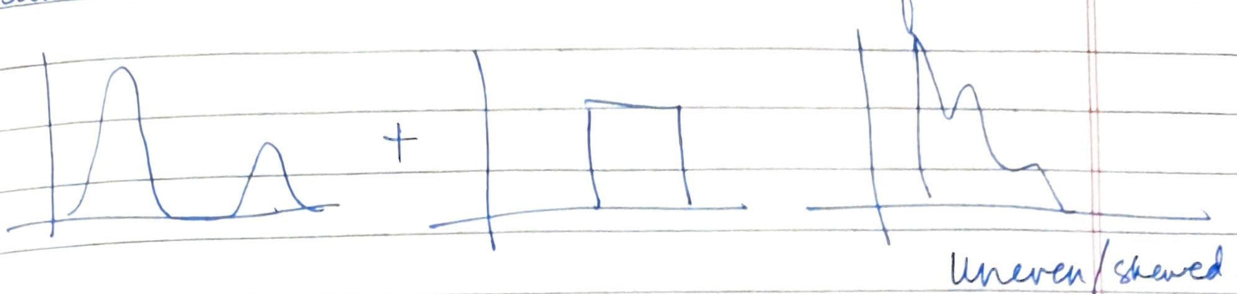
How to find T : explore the intensity histograms
Good initial first step.

Objective is to find T.



~ With increase in a lot of noise, we cannot distinguish b/w the foreground & background.

~ Same is the issue with illumination & reflectance.



Can't identify fg & bg from this as we go from L to R.

→ Thresholding forms a pipeline in many processes.

- **Global Thresh Iterative:**
  1. Select an initial estimate as T.
  2. Segment image using T.
     - pixels G_1 & G_2 : 2 groups.
  3. Compute avg int for these 2 m_1, m_2.
  4. Set new thresh $T\_new = (m_1 + m_2)/2$.
  5. If $|T\_new - T| < eps$ stop.
  6. else set $T = T\_new$.

~ This procedure is similar to the k-means algo
Finding the mid pt. of the line joining the 2.

If intensity peaks / clusters are well separated then threshold can be found.

T is usually located at one of the valleys.
Same issues of clustering come up here too.

- Run with multiple random initializations to see what happens.

- If BG similar to ROI → problematic.

• Otsu's Method

Based on histograms.
Automatically finds optimal thres maximizing b/w class variance.

→ Prelims...

- Mean & var of intensities in an image
- Yet probabilistic / normalized histogram perspective

$$M = \sum_{j=1}^{N} j P_j .$$

- Var measures : spread around the mean :-
  characteristic : better contrast compared to low var
  Diff of int WRT mean ⇒ ... high cont

oo   oo o : High interclass variance
                  Low intraclass variance

⇒ Variance : measure of region homogeneity
useful predicate. High homogenity will be low var

Otsu : Find the threshold that <u>minimizes intra class variance</u>.

Algo:
1. compute normalized hist.
2. components as p_i : $i = 0 \ldots L-1$
3. k, $0 < k < L-1$ select thresh.
4. C_1 : set of pixel levels $[0, 1, 2 \ldots k]$
   C_2 : set of pixels $[k+1 \ldots L-1]$
5. Obtain thresh that <u>max between class var</u>

$\sigma_G^2 = \sum_0 (\mu_G - r_i)^2 P(r_i)$

$$\text{max. } \sigma_B^2(k) = P_1(k)(m_1(k) - m_G)^2 + P_2(k)(m_2(k) - m_G)^2$$

$P_1(k) = \sum_{i=1}^{k} p_i$ — prob ass with class 1 & 2

$m_1(k) = \dfrac{\sum^{k} i P_i}{P_1(k)} = \dfrac{\sum r_i P(C_1|r_i) \cdot P(r_i)}{P(C_1)}$

$P_2(k) = 1 - P_1(k)$.

$= P_1(k) P_2(k) [\mu_1(k) - \mu_2 \ldots$

dist b/w means

Picking a value of k such that m_1 & m_2 are equidistant from m_G.

m_G : fixed → avg. of entire image.

If so ⇒ $P_1(k) = P_2(k)$.

Compute for each value of k & for whatever k the σ_B^2 is maximized.

Otsu respects the distribution of both fg & bg ∵ it considers the var b/w the classes unlike the iterative where assn separation.

→ In the presence of noise: first you should smooth the image & then to Otsus or any method.

Dis: Otsu not good if bg is non uniform illumination bg. ∴ No single thresh may work.

Reason: Global thresholding [won't work obv]

Sol: Subdivide image into multiple parts & use adaptive thresh / var thresh.

assm: int dist within each div is good.

Can be extended to computing σ_{xy}, m_{xy} for neighbourhood

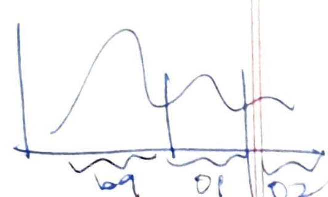
per pixel VT

$$g(x,y) = \begin{cases} 1 & f(x,y) > T_{xy} \\ 0 & f(x,y) \le T_{xy} \end{cases}$$

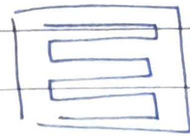
$T_{xy} = a\sigma_{xy} + b\,m_{xy}$ or $+ b\,m_G$.

func. of neigh stats.

Dual thresh ←



→ can also compute image of local std. deviations

• **Moving Average:** avoid any sys bias  → processing in zigzag manner

Streaming data $x_1, x_2 \ldots x_N, x_{N+1}$

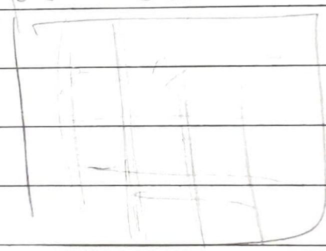
$$\mu_{N+1} = \frac{x_1 + x_2 + \ldots + x_{N+1}}{N+1} \leftarrow \text{window size}$$

$$\mu_{N+1} = \frac{\mu_N N + x_{N+1}}{N+1}$$

Replace pixel value with moving avg.
Works v. well with doc. images.
→ lighting source causes spot shading

To apply local thresholding choose a window size for the same.
Window $<<<$ text. Fairly common in scanner type applications

Illumination variation



sinusoidal shading

→ Moving avging works well here

Takeaway

size of ROI / objects
small : adaptive
large : global

• **Region Based Segmentation**

→ Start with seed of points (like flood fill)
Region growing : find the common predicate for the seed point. Some func. of RGB vals
$|s_R - C_R|$ $|s_G - C_G|$ etc.

→ Region splitting & Merging
Define splitting & merging functions
Predicate (O_1, O_2) ... decides whether 2 segments to be given same label or not.

→ Clustering :-
We clustered based on colours earlier
Seg related to neighbourhood. (agglomerative, spectral etc)
∴ include spatial coords.

∴ We cluster based on colours + spat.
feature vect : $[r, g, b, x, y]^T$

• Humans Methods :- Gestalt principle
1. Similarity.
2. Proximity.
3. Closure
4. Continuation
5. Common fate.
6. Symmetry
7. Parallelism
8. Familiarity

x ———————— x

• LECTURE 19: RESTORATION :-

Some kind of degradation of image :- shook camera, inherent bg, quality of sensors or quantization issues.

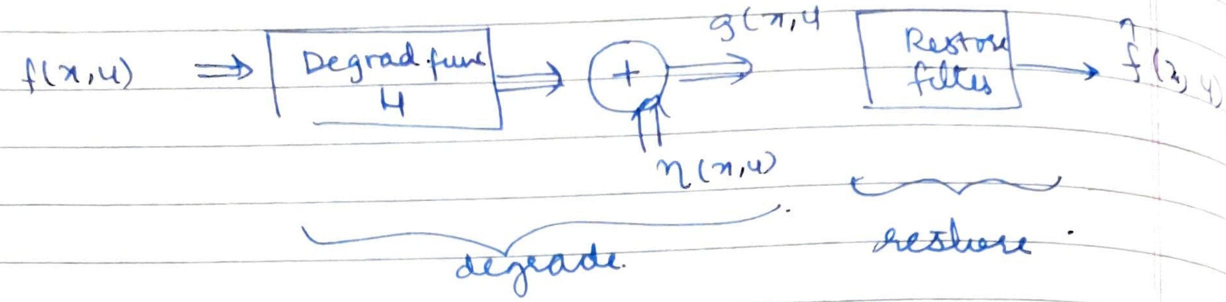
→ Light field cameras :- capturing volume of light can then be sliced acc to different focii

→ Camera shake, Haze removal, blur focus etc.

Enhancement : subjective
restoration : objective → original

→ Image restoration → photometric (intensity)
→ geometric

- Model :-



→ $g(x,y) = H[f(x,y)] + \eta(x,y)$.
$\hat{f}(x,y)$ should be close to $f(x,y)$. Design some kind of a restored dist metric.

- Use apriori knowledge of degradation (known)
- Model degradation Apply H^{-1} reverse
- Formulate & evaluate

2 parts :- → Degrad func H : Linear & Pos invariant
→ Noise : Indep of spatial loc & uncorrelated with image
Noise is not a func. of intensity of image.

→ If H is linear & pos invariant can be modelled
$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y)$
action of ~~H~~ h can be modelled as a conv of h & f.
∴ From fourier transform :-
$G(u,v) = H(u,v)\,F(u,v) + N(u,v)$.
Can be obtained in the fourier domain

Step 1: Assume image degraded only by noise
H identity ∴ $g(x,y) = f(x,y) + \eta(x,y)$

Date
Page
classmate

Now ∃ diff types of noise models that can be applied.

- Eg : $p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(z-\bar{z})^2}{2\sigma^2}}$ ... very convinient.

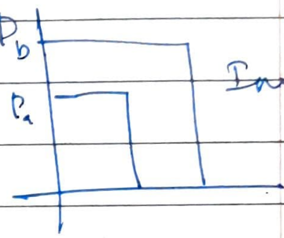
- Rayleigh Noise : [Radar velocity images] Corrupted by this. (useful for skewed histograms.

$$p(z) = \begin{cases} \frac{2}{b}(z-a)\, e^{\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a. \end{cases}$$

$$\bar{z} = a + \sqrt{\pi b / 4} \qquad \sigma^2 = \frac{b(4-\pi)}{4}$$

- Earlang (Gamma Noise) : Images from lazer.
- Exponential noise.
- Uniform noise. (quantization, most unbiased) defined over our dist. of intensities.

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



Impulse.

salt & pepper noise

$P_a = P_b \Rightarrow$ unipolar noise

Impulse → sync in digitization, sensor malfunction.

→ Not v. easy to identify noise model purely from appearance.

How to study system noise:

a) Noise calibration : capture 'flat environments', MLE.

b) Select model with better stat test scores AIC, LRT etc.

Without access to cameras & all,

- estimate from patches of const. intensity
- For impulse noise :- use mid-gray patch / area.

- <u>Restoration by:</u>

→ Arithmetic filter $\hat{f}(x,y) = \frac{1}{mn} \sum g(s,t)$.

Trade off : small ~~noise~~ blurring

→ Geo mean $= \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$.

GM : preserves the detailed structure better than AM. but expensive.

→ Harmonic : $\hat{f}(x,y) = \dfrac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$

good for salt but fails for pepper

→ Contraharmonic : $= \dfrac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q}}$

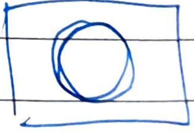
$Q > 0$ pepper $Q < 0$ salt

AM $Q = 0$ HM $Q = -1$

→ Spatially varying noise

To remove sinusoidally varying noise & all, use band pass/reject filter.

removes grid noise



- <u>Estimating Degrad Func:</u>

$g = h * f + n$

∃ three main ways to do this :-

→ Observation : look, find & iterate [match the 2]

→ Experimentation : important idea for calliberation

→ Math modelling.

If noise is absent degradation func ⇒ scaled impulse func.

classmate Date Page

→ Motion Blur :-

- Exposure
- If amt. of light hitting the sensor changes significantly over exposure period → Motion Blur
- Person, Camera moves, change in lighting.

Filter $\begin{bmatrix} I \end{bmatrix}$ on image get a motion blurring image.

Can assume linearity of motion.

coords come from some other loc. also → $g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]\, dt$. T: period of motion

$$G(u,v) = F(u,v) \int_0^T \underbrace{e^{-j2\pi[u x_0(t) + v y_0(t)]}\, dt}_{H(u,v)}$$

uniform motion blurring: If $x_0 = \frac{at}{T}$ & $y_0 = \frac{bt}{T}$: uniform velocity

the blurring can be modelled.

→ Atmospheric Turbulence

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

- Recovering : even if wkt degradation func, we cannot get the undegraded image

~~If no noise.~~

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

sub from main :- ~~F~~ $\hat{F} = F + \frac{N}{H}$

① - Problem $N(u,v)$ is random func → cannot compute FT.

② - If degradation has 0 or small vals $\frac{N(u,v)}{H(u,v)}$ dominates

ie: $\hat{F}$ will be all 255s.

problem $g(x,y) \xrightarrow{F} \frac{G(u,v)}{H(u,v)} = \hat{F}(u,v) \xrightarrow{IF} \hat{f}(x,y)$

INVERSE FILTERING.

∴, given original image. other option is to use <u>Weiner filter</u>

Assn : noise & image uncorrelated.
consider image & noise as RVs.

$\boxed{\text{min}}$
∴, $e^2 = E\{(f-\hat{f})^2\}$ estimate avg. error.

At value of min of error func.

∴, $\hat{F}(u,v) = \left[\frac{H^*(u,v)}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \underbrace{S_\eta(u/v)/S_f(u,v)}_{*}} \right] G(u,v)$

stable version of IF.

power spec of degradation func

Explicity accounting for power spectrum of 2 things

$S_n(u,v) = |N(u,v)|^2$ = noise (auto correl of noise)

$S_f(u,v) = |F(u,v)|^2$ = undegraded image

Chicken & Egg

∴, we approx * with K

Power spectrum :- Taking fourier & its magnitude

23/10/20

LECTURE 20.

- Math model.
  - H : Linear & Pos invariant
  - Noise is decoupled.

$$g(x,y) = H[f(x,y)] + \eta(x,y).$$

→ when H is Identity only exclude noise.
→ Diff times of noise.

→ . To study noise model : do noise calibration.
Capture flat environments.
- select model with better statistical score.

- If only img available.
  - use estimate from patches of intensity

→ Restore in presence of only noise.

AM : $\hat{f}(x,y) = \frac{1}{mn} \sum g(s,t).$

GM

HM : $\hat{f}(x,y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}.$

→ Countraharmonic mean filter. (General purpose)
- Q : order of filter.
- Good for salt & ppeper
- Elim. pepper Q > 0 salt Q < 0.
- Q = 0 AM ; Q = −1 : HM.

Eg: non-linear filters : max - min etc.

→ Periodic noise can be eliminated using band pass

classmate
Date
Page

How to model $H(u,v)$.

Motion blur.
$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin[\pi(ua+vb)]\, e^{-j\pi(ua+bv)}$$

Recovering the image in more complicated settings
then:- estimate
$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}.$$

- Recovering img in presence of noise + deg is not possible.

- This where Weiner filter can be used to perform the inverse.

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v).$$

param accounts for noise

→ what exactly is unknown in the math problem
$$g(x,y) = H[f(x,y)] + \eta(x,y).$$

	Known / Given	Problem
	H, g	Recovery.
	g	Blind recovery.
∈ particular family g,	H (partially)	Semi blind rec
model H:	f, g	System identification.

All the above are called inverse problems.

All this is distortion due to intensities

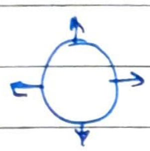
CLASSMATE
Date ______
Page ______

Geometric Distortion :

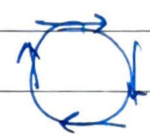
Pinhole camera :- image a real world object.

- → Chromatic abberation :- Lights of diff wavelengths dont converge at same point.
- → Radial & Tangential distortion.
- → Vignetting : don't uniformly receive light from all sources

Lens glare : starry interreflections of light within optical lens system.



Radial.



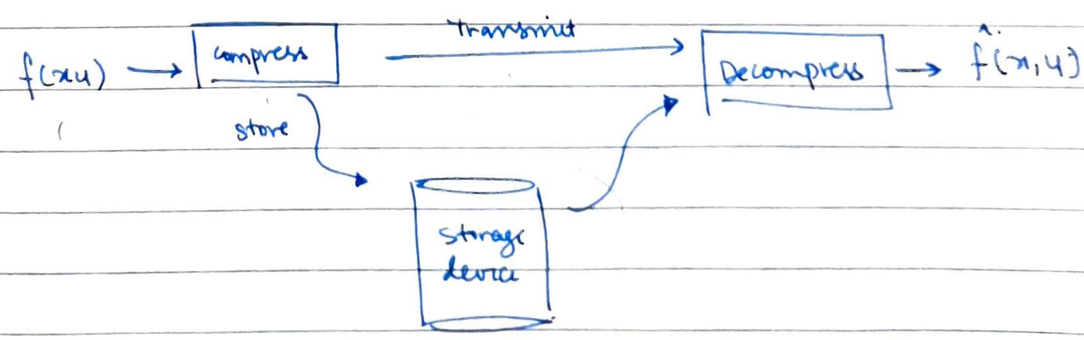
Tangential

Rectified using geometric camera calibration.

IMAGE COMPRESSION :

Aims to reduce the amount of data while preserving as much info as possible.

~~Rectifier~~



- Transmit : stream across some network or something
- Obtain more or less same image

can be done by exploiting redundancies in image data.

- Compression ratio:
  extent to which we are able to reduce amt of data rep.

$\boxed{\text{Data set image}} \Longrightarrow \boxed{\text{Data set 2 compressed}}$

n_1 carrying units — n_2 carrying units.

> ratio ⇒ better conversions

COmp Ratio : $C_R = \frac{n_1}{n_2}$.

2 forms: uncomp : comp. :: $n_1 : n_2$

↓ the n_2, better is the compression achieved.

- Data Redundancy.

$$R_D = 1 - \frac{1}{C_R}.$$

$C_R = \frac{10}{1}$ then $R_D = 1 - \frac{1}{10} = 0.9$.

⇒ 90% of the data in Dataset 1 is redundant.

To put it on a 25GB blu ray disc, required compression factor = 53.6.

→ Types Of Redundancye:

① Coding Redundancy :-
Code is a list of symbols
Code Word :- bunch of symbols used to rep some info
Length : no: of symbols used.

Terminology :-

$N \times M$ image

r_k : k^{th} gray level.

$l(r_k)$: # of bits of r_k.

$P(r_k)$: prob :- histogram

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

$$\text{Total} = NM.L_{avg}.$$

Case 1: constant length code.

If constant length ⇒ 3 bits required.

Case 2: variable length encoding

L_{avg} here is < 3 ie: 2.7 bits !

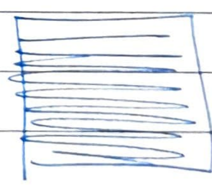
Now ~~#~~ bits :- 2.7 NM bits

$$C_R = \frac{3}{2.7} = 1.11. \qquad R_D = 0.099.$$

② Spatial / Temporal Redundancy.

∃ relationship with prob & code word.

↑ prob :- smaller length code words.



Here everything is occurring frequently ⇒ cannot reduce code length.

Do a run-length encoding of the image.

Represent each row by : (i, n).

i : int, n : no: of pixels.

Nice!

Eg : (20, 256) ... (170, 256) .... etc.

∴, 16 bits / row.

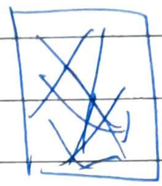
⇒ 16× 265 bits.

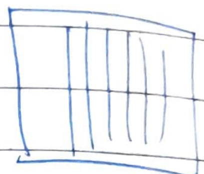
ie: $\dfrac{256 \times 256 \times 8}{16 \times 256} = 128 : 1$. Compression

ie:- (105, 20), (60, 4)....
eg: documents / uniform patches.

whenever $\exists$ interpixel redundancy : pixel values correlated
ie : pixel can be reasonably predicted by neighbours.

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f(x)\, g(x+a)\, da$$


no


yes

auto correl $\Rightarrow f(x) = g(x)$.

- with repeatative structure, auto correlate values $\uparrow$
- How self similar is the image.

- can apply run-length encoding ~~horizontally~~ vertically $\downarrow$
- autorellation : pre test for whether RLE can be applied

At times we may have to transform the image to obtain the interpixel ka value relation.
→ threshold & perform RLE eg: (1+10). bits / pair
0 or 1 for binary — for count

convert to diff domain

Temporal Redundance :-

Frame t, Frame t+1.
∴, Store frame(t+1) − fram(t) and store.

RLE (δt).
↓
residual

$(n_w + n_h + 8)$ or $H \times W \times 8$

Date
Page

③ Psychovisual Redundancy :
exploit the limitations of HVS. (peculiarities).
throw info considered same by eye.

COMPRESSION CONTD :

1/10/20

Goal : reduce amt. of data req. to represent digital image.

$CR = \frac{n_1}{n_2}$: n_1 : before n_2 : after

Info $= -\log_2 p_i$ bits $E = -\sum_{i=1}^{L-1} p_i \log p_i$

cannot do any better than entropy

Psychovisuality : After quantizing with fewer bits, there may exist some discontinuities ∴, can add add small noise : pseudo random no:s to image

I.C Model :-

$f(x,y)$ → Mapper → Quantizer → Symbol encoder → channel

Mapper: reversible; Quantizer: not reversible; Symbol encoder: reversible

- Mapper : transform data to account for interpixel redundancies
- Quantizer : quantize to account for psychovis red.
- Encoder : acc for coding redundancies.

Reverse :

Channel → Symbol decoder → Dequantizer → Inverse mapper → $\hat{f}(x,y)$

Dequantizer: irreversible in general.

→ As long as $\hat{f}(x,y)$ looks close to $f(x,y)$ or it is good enough for image processing tasks we are good.

→ If $\hat{f}(x,y)$ is ↓ in dim or ~~width~~ bit depth : more efficient representation $f(x,y)$ → processing saving

classmate

Date
Page

- **Lossless Compression**

$f(x,y) \longrightarrow$ Compress $\longrightarrow$ Decompress $\longrightarrow \hat{f}(x,y)$

$$e(x,y) = \hat{f}(x,y) - f(x,y) = 0$$

Achieves perfect compression.

Lossless Techniques

- R.L.E
- Stat encoding
  - Huffman
  - Arithmetic
  - LZW
- Lossless Predictive coding
- Bitplane encoding

- **Huffman Encoding**: var length coding tech.

→ Source symbol encoded one at a time 1-1 map

→ Optimal code

∃ 2 passes

Forward pass :-

Desc order [compute normalized histogram ∴ probabilities

Combine last 2 prob to make a single.

Repeat until only 2 prob remain

step 1: Is to form this sort of a prob tree

Eg: $\underbrace{01010}_{a_3}$ $\underbrace{011}_{a_1}$, $\underbrace{1}\underbrace{1}\underbrace{1}\underbrace{1}$ $\underbrace{100}$
$a_2\ a_2\ a_6$.

To encode, assign code symbols backward

Last 2 probs :→ 0 , 1.

then append 0, 1 to remaining, in the same manner (keep track of intensity value for corresponding vals

Weakness of huffman : encodes one at a time

→ Decoding can be done unambiguously using a lookup table.

- **Arithmetic encoding** : or (Range Enc)

Addresses bunching & encoding sequences of source symbols together.

- No 1-1 mapping
- Slower than huffman but can achieve better compression

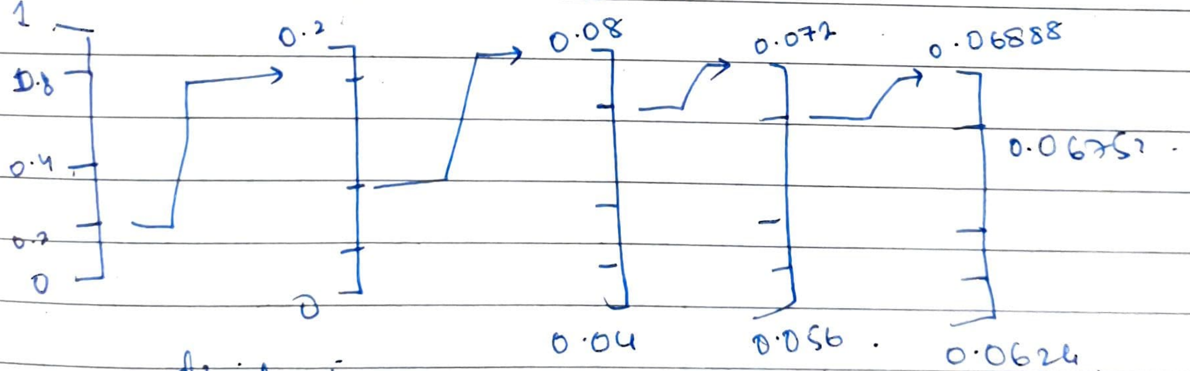
eg:

a_1	0.2
a_2	0.2
a_3	0.4
a_4	0.2

Encode $a_1 a_2 a_3 a_3 a_4$

map to regions of prob

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, 1.0)



divide in same values.

consider respective subintervals

∴ Resultant interval [0.067552, 0.0688)

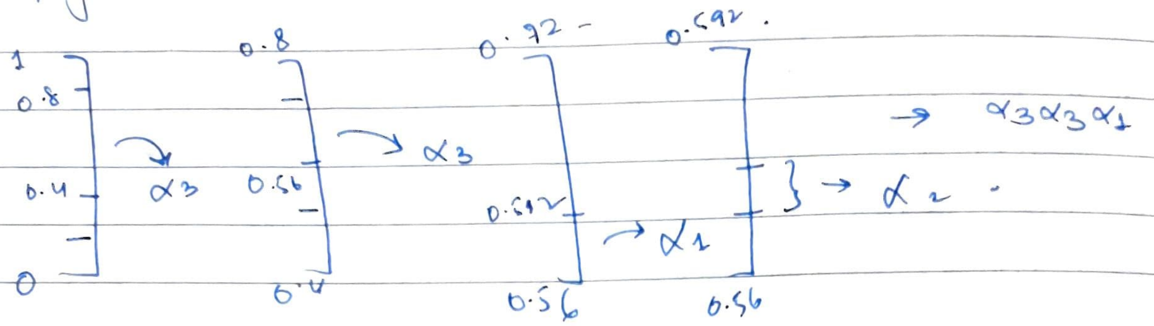
∴ Arithmetic encoding of symbol ≐ 0.0688

→ The sub interval required to represent the msg ~~which~~ becomes smaller & smaller as the no: of symbols in the msg increases.

Decoding :-

Given ~~0.0~~ 0.572 & code length = 4

Again have access to probability table.



- LZW coding :-

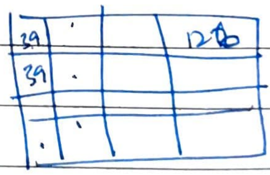
- Requires no prior know of symbol probabilities
- Assigns fixed length code words to var length symbol seq. No 1-1 correspondes.
Used in GIF, TIFF PDF.

→ codebook | dict must be constructed (data dependent manner. auxiliary table).

scanning image : row by row

Whatever gray level seq. not in dict, assigned to new entry.

0	0
1	1
⋮	⋮
255	255
256	?



39 yes

39-39 no ∴ .... add 39-39 → 256

39-126 ∴ add 39-126

- Dict need not be sent with encoded input
- Can be built on the fly.

why consider only certain chunks

CR = empty
repeat
P = next pixel
CS = CR + P

if CS is found
- no out
- CR = CS
else
Out D(CR)
Ad CS to D
CR = P

→ RLE is good but cannot apply to not nice images. Sometimes they get bigger
can compress any type of data × high ratios.

Can now mix & match these images.

→ Bit Plane Coding :-
Process each bit plane individually.
a) decompose to series of bin
b) Compress (using RLE.)

- Lossy Methods :

Block Truncation coding	Lossy Predictive coding	Transform Coding	Subband coding	Fractal Coding	Vect Quant

$$e(x,y) = |\hat{f}(x,y) - f(x,y)| > 0$$

→ Transform image to some other domain to red. inter pixel redundancy.

Eg: throw away the higher freq

$$\frac{1}{N} \sum_{u=1}^{K-1} \sum_{v=1}^{K-1} T(u,v) \; h(x,y,u,v)$$

T(u,v) can be computed using DFT, DCT, KLT ... PCA
JPEG uses DCT

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

One diff: no complex parts.
Entirely defined in terms of cosines.

WHT: Walsh Hadamard function: uses 0-1 basis ~~function~~

- Take img & divide into sub patches.
- May be better to represent patches in compressed format better

DFT prone to artifacts at edges.

Procedure:

- Divide image into 8/8 patches
- For each patch encode using DCT.
- Remove 50% of co-efficients in transform domain

Taking RMSE, DCT is better. DCT < WHT < DFT
↳ encodes items at patch level better. Causes fewer artifacts at a patch level

Finding: 8x8 images are better.

- **JPEG Compression:**

Image divided into 8x8 → [Forward DCT] → [Quantizer] → [Entropy encoder] → Com

& shift gray levels to -127 → 127

[-128, 127] compatibility with DCT.

→ Quantization table $C_q(u,v)$ round = $\frac{C(u,v)}{Q(u,v)}$

1 < qual < 25 best compr

DCT coeff

Given quality factor, Higher freq. entries made 0. bottom right matrix.

→ Order coefficients in zig-zag manner
∵ We can use RLE

↑ Quality factor, Compression is vv high but little noise may come in

JPEG insufficient for things like fingerprint. FBI uses WSQ. No "blocky" artifacts therefore.

→ Final step in JPEG:-
Instead of using RGB it uses Y Cb Cr & performs chroma subsampling: quantizes the Cb Cr aggressively. diff quant table.

JPEG supports diff modes:
- sequential: LR TB
- For web based we use progressive (multi resolution image). Lower freq. sent first

Eg: GIF: lossy PNG: lossless
MPEG: exploits temporal redundancy.

03/11/20

- **Comparing Images:**

- Match patterns across image.
- Similar version in HAM transforms s.t. struct el maps to image & captured.

Q: How to tell if 2 images are same.
Obv: pixel-by-pixel comparison.
Noise, quantization etc. introduces differences.
- Depends on who is answering the question.
- To define what numerical diff is acceptable

ie: $V = \frac{\Sigma(a_i - b_i)^2}{MN}$ $V < \theta \Rightarrow$ same.

All factors need to be considered.

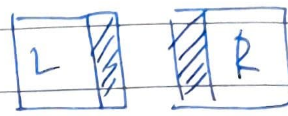
→ Better approach: template matching:

Identify similar sub images within 2 images.

Applications:

- Match R & L pic of stereo image.
- Find particular pattern in scene.
- Track moving patterns (suspect's car)

Procedure forms basis of DIP & CV.



Predictive matching / block matching as done in video frames → similar here.

- Basic idea.
  - Move given pattern over search image
  - Measure diff b/w template & sub images
  - Record pos where diff ↓ & sim ↑
  - Output is a set of locations

- Problems:
  - What is distance / diff measure.
  - What level of diff considered match.
  - How results change when brightness / contrast change (diff lighting conditions to be accounted for).

Score for each (r, s) plotted

eg measures:-

- Sum of absolute diff :- $\sum_{i,j \in R} |I(r+i, s+j) - R(i,j)|$
- Max of all such diff.
- Squared differences (N-dim Euclidean dist)

Output from template matching can either be top left corner or origin of the patch

classmate
Date
Page

Squared distance :-

$$d^2_E(r,s) = \sum_{i,j \in R} (I(r+i, s+j) - R(i,j))^2$$

$$= \underbrace{\sum I^2(r+i, s+j)}_{A(r,s)} + \underbrace{\sum R^2(i,j)}_{B} - 2 \underbrace{\sum I(r+i, s+j) \cdot R(i,j)}_{C(r,s)}$$

B: const term ∴ can be ignored

like a similarity measure

$C(r,s) \rightarrow$ linear cross correlation b/w I & R

sliding template across image for diff shifts.

$$(I \circledast R)(r,s) = \sum \sum I(r+i, s+j) R(i,j).$$

Considered 0 outside bounds ∴ $\sum_{i,j \in R}$ only.

Now if A is constant, no matter where you consider a patch of template size :-

☆ Min value of $d^2_E(r,s)$ corresponds to max value of $(I \circledast R)(r,s)$.

→ Range of $C(r,s)$ not bounded, ~~bounded~~ depends on template size

Not invariant to changes in intensity amplitude.

- Norm cross Corellation

$$C_N(r,s) = \frac{\sum I(r+i, s+j) \cdot R(i,j)}{[\sum I^2(r+i, s+j)]^{1/2} \; [R^2(i,j)]^{1/2}}.$$

$C_N(r,s)$ is a local distance measure [0,1] range

$C_N(r,s) = 1$ indicates max match.

$C_N(r,s) = 0$ indicates indicates images very dissimilar

⇒ Score indicates level of matching

∴ Can set a threshold b/w [0,1].

→ Sum of abs diff : okay but affected by global int changes
→ Max diff : Responds more to lighting than pattern
→ Corr coeff

→ Variant : zero normalized cross correlation
Range : [-1, 1]
Subtract $\bar{I}$ $\bar{R}$ from the num & deno qtys. ∴ 0 mean.

$$\bar{I} = \frac{1}{K} \sum I(r+i, j+s) \qquad \bar{R} = \frac{1}{K} \sum R(i,j)$$

pixels in R

$$S = ZNCC(I, R) \ldots \text{matrix}$$

→ ~~Sum of absolute differences :- affected by global intensity changes.~~ sq. euclidean dist. ?

→ Ideal image :- only maximas are higgghhhh

Correlation coefficient produces the best results ideally. robust measure in realistic lighting conditions

To obtained diff configurations :- try variants of the template ← rotate 90° etc, scale etc

If we get [-1, 1] ....
-1: opposite, exactly negative
1: exact match

→ Template need not be rectangular.
→ Can use circular, elliptical, or custom shaped
→ Weight matrix stored in a matrix while pattern matching. can weigh in diff values.

→ Matching & Rotation : computationally prohibitive
[Logarithmic polar space matching
Affine matching : do not want to match euclidian template. Can transform & use]

Find stable regions in affine mapping.

- <u>Binary Images</u>:

→ ~~Sxx~~ Some missing pixels are okay. Output nice
problems Very small rotations / distortions ⇒ large distances
∴ need more tolerant measure.

To account for things :-
Use distance transform :- distance to closest fg pixel

→ <u>Chamfer matching</u>: template matching for binary images. :→ chamfer score: avg nearest dist from template points to image points.

ie: wherever it matches best ⇒ dist least [from dist image]
→ computationally inexpensive

- Dist image provides a smooth cost function. And as we move away from matching loc, score increases in smooth manner.

Score :- ① for each loc of template ~~look at the~~
② look at dist transform values

$$Q = \frac{1}{K} \sum_{i,j \in (R)FG.} D(r+i, s+j).$$

only for the i,j that ∈ to foreground in R template
add it to the ~~xx~~ D.

Zero Q score ⟶ max match
Large Q score ⟶ large deviation

Chamfer
Variant to scale & rot
Sensitive to small shape changes
Large no: of template

Robust to clutter
comp cheap

→ Build sort of classifier : cascade based obj detection
→ shape contexts

Reducing Cost of Correlation:

→ large costs of correlation matching.

→ face recognition] every single thing may
char recog] be too much.

→ matching at the right scale may also be a cause of concern.

→ Result for single scale search :- can sometimes result in false positives.

→ cascade based object detection algorithm.

6/11/20

• **REP & DESC:** (end product of many img proc pipelines)

→ Low-level img analysis.
preprocessing, extracting primitives.

→ Mid level :- img rep + stat. description
task spec. feature design. +ML.

→ Deep learning : obj understng. to image und.
wild data :)

• Segmentation, morph operations to define as
- lines - circles - points etc.
∃ all in pixel level

understanding

pixels → Regions points lines → extracting primitives → Eng / lang description

- Representation : Image with one or more objects.
- Encoding of object : can be a 1D vector (feature vec)
- vector to function as proxy : truthful but approx.

- Descriptor :
- Only aspect of the object (Eg: its length, no: of wheels)
- suitable for classification
- Have desc. invariant to noise/translation

Image analysis $\xrightarrow{rep}$ Internal
$\xrightarrow[rep]{}$ External → shape or ~~boundary~~ of regions.

→ Ext Rep :- Rep in terms of end coods / polygonal.
Ext desc .- perimeter, area anything

→ Internal rep : Region : ~~Reps~~
Rep : pixels inside the region.
Desc : Colour, Texture etc.

Eg: Understanding form filling :
- Identifying regions & all external rep
- To understand who has filled what, diff b/w text & handwriting → internal rep.

Defining an Object :- All 4 coods or one cood & ~~Ed~~ ht & wd (external descriptors.)

→ Semantic segmentation : to identify class of objs.
Instance seg : to also count them & IDs.

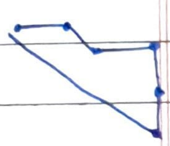
- Boundary rep set of pixels.

- Boundary Following Algo :
Given a binary region R or its boundary

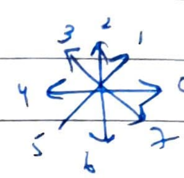
1. Let the top left be starting point b_0. label 1
2. Denote c_0 the west neigh of b_0. c_0 : bg
3. Examine 8 neigh of b_0 (starting at c_0) Let b_1 be the first 1 pixel

Repeat ....

→ Boundary description :-
Chain Code (Freeman)
Boundary : A connected seq of straight line segments of specified length & direction.



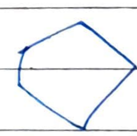
Encode as : 0, 7, 0, 6, 6, 3



Initialization depends.

→ Polygonal approx.

Concave



Convex

Minimum perimeter polygon : stretch rubber band

10/11/20

Digital Video Processing :

- First moving pic were on film.
- Because of bet
- Does horse have all 4 hooves off ground.

→ Muybridge experiment :- cameras fire for each slot. Standford won the bet & all 4 hooves were off.

→ Viewing these photos really fast.

→ Film is an analog medium but its discrete in time.
[ ]

→ TV is a technology for transmission & reprod of moving picture.

→ Rasterisation : allows images to be converted into 1D signals for transmission. [signals continuous → dis ↑

Signal represented in Y U V format instead of RGB.

sin cosine change tint in colour

NTSC composite vid :- $c = Y + U\sin(\omega t + 33^\circ) + V\cos(\omega t + 33^\circ)$

→ Storing TV signals on magnetic tape
Latest era : HD DVD, Blu-Ray, HDV.

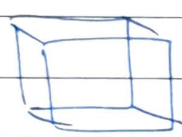
→ Progressive Scanning
→ Interlaced :- First draw odd lines then go back & draw even lines.

V. diff to explain still frames from TV.

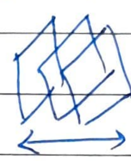
Digital video capture :- CCD array capture
Variant of JPEG used for video compression.

→ CCD vs CMOS ... cheaper.
CMOS use 'rolling shutter'.
- Sensor exposed line by line (may be prob.)
- wavy looking image

→ HVS extremely sensitive to motion. Capturing motion in fMRI ... interested in studying brain function.
- Stack of frames in a video.
- Take video & think as certain groups of frames.



24|36|52
channel.



apply on these individual frame.

- Gait based biometrics.

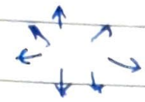
• Diversity of Motion :-
  - Time scale
  - Localized? specific locations?

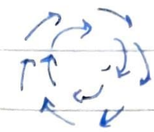
Predict how camera moved in 3D space.

→ Time varying image formation model.
2 side : geometric & photometry
movement & struct   appearance

→ Movie editing :-
camera panning, zoom in & out, combination.

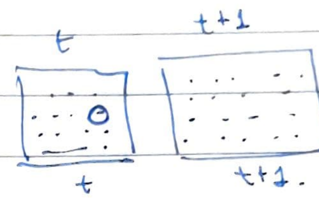
→ Camera Motion:

a) Camera zoom  → Optical flow

b) Camera rot around z-axis (roll.) 

c) Flow view: if ∃ some intensity at some loc (x,y) at time t, ∵ we typically have coherent motion

$I_t(x,y)$ goes where $I_{t+1}(x,y)$.



- Use flow vector to characterize this bunch of arrows.
- Looking at flow field we can say about the motion.

ie: $I_t + O_f \longrightarrow I_{t+1}$.

• Real World Motion:

→ Rigid motion vs deformable motion.

→ Overlapping vs. non overlapping

Optical flow can be used as a feature representation or descriptions (raw info).

Form histogram over θ.

→ Along with RGB we want motion vectors as well.

Video acquisition → camera / Computer gen videos.

→ Removing unwanted tracking marks watermarks by video inpainting. Exploit temporal nature of video

Use values from surrounding pixels to change.

1) ~~Use~~ Isolate the text.
2) Then remove.

→ Blocky artifacts Error concealments
Removing info

→ Superresolution :- LR seq → HR seq.
intelligent upsampling

• Background subtraction :- identify fg objects in that image.

Algo ~~should~~ handle :
- sudden / gradual illumination changes.
- High freq rep. motion in bg.
- Long term scene changes.

pixels ⟶ semantics

Thresholding frame I_t, I_{t-1}, pixelwise thresh.

$$|I(x,y,t) - I(x,y,t-1)| > Th.$$

Somewhat better way is to use mean filter.
Estimate bg as mean of prev frames.
ie: Obj not present all the time

Obj captured & stretched across.

Median fairly robust & gives better results.

→ If frames changing faster α will be higher