INDEX

Name Souradip Nath 8th Semester

Std. Sec. Roll No.

Sub. Year.

Sl. No.	Date	Page No.	Title	Teacher's Remarks
			Digital Image Processing and Pattern Recognition	
			Prof Santi Prasad Maity	
			Rafael C Gonzalez	
			Richard E Wood	

13th Jan, 2022

Digital Computers are much efficient processing numeric data. But, the information that we mostly gather from the nature is non-numeric in nature. So, can the machine be superior than human in processing non-numeric data?

This gives rise to the need of having a high-IQ machine

→ What do you mean by "Image"?

Image is a signal ie a function ie a mapping between light intensity from pixels.

Mathematically, Image is nothing but $f(x,y)$
where, (x,y) is a co-ordinate pair corresponds to points known as pixels. f maps pixels to light intensity.

If both (x,y) and f are discrete and finite, then we call it a Digital Image

→ Why do we need "Processing" of Image?

(i) Improvement of pictorial information for "human interpretation"
(ii) Improvement of pictorial information for storage, retrieval, and representation from the machine perception → "Autonomous Processing"

→ What are the basic operations involved?

"Sampling" and "Quantization" to convert Analog Image to Digital Image

- Improving the visual quality of an Image → Image Enhancement
  → Improve the Brightness, Contrast etc
  - Spatial Domain Approaches
  - Transform Domain Approaches
- Image Restoration

"Spatial Domain Processing" → The pixels values are manipulated directly in the spatial domain

"Transform Domain Processing" → Here, a fourier transformation is applied to transform the domain to frequency domain. Then, the pixel values are manipulated there. The filters co-efficients are multiplied. Then, inverse fourier transform is applied, and the image is observed in Spatial Domain.

Digital Image Types →

(i) Intensity Image or Monochrome Image 0-255
(ii) Color Image or RGB Image (0-255)×3
(iii) Binary Image or B/W Image 1-White, 0-Black
(iv) Index Image

Image Acquisition Process →

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

20th Jan, 2022

- $i(x,y)$ → Light Intensity
- Illumination → $[0, \infty)$
- $r(x,y)$ → Reflectance → $[0, 1]$

Generating a Digital Image → The two processes that are required to generate the Digital Image from the continuous analog image are, Sampling, Quantization

* The sampling rate dictates the details of the picture
  - ↳ Nyquist Rate comes into the consideration

* For comparing Spatial Resolution, we must have same Field of View ie the same area must be covered by the pixels whatever be the number

The discernable detail that is represented by a small pixel is known as the Spatial Resolution

Single Pixel Area is large ← Sampling Rate is less
↳ Number of Pixels covering the Field of View is less
↳ Poor Spatial Resolution ≅ "FoV Constant"

But, along with the undersampling, if we keep on reducing the field of view proportionately, the area covered by a single pixel remains same. So, no change in Spatial Resolution.

≅ "FoV Not Constant"

Image Quantization → $2^{N_v} \simeq N_c$ → Number of Color Levels

N_v : Number of bits representing Color Levels

As we keep on decreasing N_v values, the color depth is reduced
Low Color Depth gives rise to **False Contour**

Representation of Digital Image → 2D Matrix of Light Intensity Values

And the direction of axis is, ↓ x, → y

4-neighborhood relation considers only "Vertical" and "Horizontal" Neighbors.

8-neighborhood relation considers
- 2 Vertically Up down
- 2 Horizontally Left Right
- 4 Diagonal

Diagonal-neighborhood relation contains only 4 diagonal

IMAGE ACQUISITION

The process of sensing our surroundings and the representing the measurements in the form of an image.

Passive Imaging → Source of Illumination is Sun
Active Imaging → External Lighting Source

Histogram Statistics

R is the discrete random variable representing the image

$$\mu_n(R) = \sum_{i=0}^{L-1} (R_i - m)^n P_{R_i}$$

n^{th} central moment
m: mean
P_{R_i}: Prob(R_i)
0, 1, ..., L-1 : Intensity Values

$\mu_0(R) = 1$

$\mu_1(R) = 0$

$\mu_2(R)$ = Variance

14th Feb, 2022

Noise Cleaning

(i) Noise and the signal are uncorrelated for each location
(ii) Noise should have zero mean → Assumptions about the noise

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

Image Averaging: $\bar{g}(x,y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x,y)$

K copies of noisy images
K = 8, 16, 32, ...

$$E[\bar{g}(x,y)] = f(x,y)$$

Expectation of the average of the noisy image produces the original image

* More images, better the produced image.

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)}$$

Averaging results in reduction of Noise variance

For a Single Noisy Image, we have to perform the averaging operation on the neighbowehood pixels of individual pixels. We can varey the window size, to decide which neighbowehood pixels to include.

Spatial Filtering → Sometime we need to manipulate values obtained from neighboreing pixels

For example, how can we compute an average value of pixels in a 3×3 region centere at a pixel z?

Step 1. Selected only needed pixels (3×3) → Mask or Window or Template

Step 2. Multiply every pixel by 1/9 and then sum up values

3	4	4
9	7	6
3	6	1

Neighborehood

× 1/9

1	1	1
1	1	1
1	1	1

Mask/Window

$= y = \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 4$
$\vdots$
$+ \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 1$

"Low Pass Filter"

Now, fore computing the 3×3 avereage values at everey pixels, we imagine that we have a 3×3 window everey where on the image.

Step 3. Place the result at the pixel in the output image

Step 4. Move the window to the next Location and go to Step 2

Main Concept

① Move the reference point (center) of mask to the location to be computed

② Compute **sum of products** between **mask coefficients** and **pixels in subimage** under the mask

Mean Filters are Linear Filters

How to Design the Mean Window?

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Equal Weighted Filters

(i) Square Window

	1/5	
1/5	1/5	1/5
	1/5	

(ii) Plus Shaped Window

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Unequal Weighted Filters

(iii) Weighted Mean Square Window

	1/6	
1/6	1/3	1/6
	1/6	

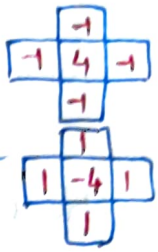
(iv) Plus Shaped Weighted Mean Window

Mode, Median, k-nearest Filtering → These are non-linear filters

Mode → A pixel value is replaced by its mostly occuring neighbour ⇒ Calculate Mode and replace

Median → A pixel value is replaced by the median of its neighbours ⇒ Calculate Median and replace

Laplacian Masks → $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$



(i) Forewared Difference : Centre of the mask is positive

(ii) Backwared " : Centre of the mask is negative

Sum of all the co-efficients must be zero

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) : \text{Backwared Difference} \\ f(x,y) + \nabla^2 f(x,y) : \text{Forewared Difference} \end{cases}$$

Output image is supposed to have Enhanced Grayish Edges on a dark background. The tonality of the background is lost, and the edges are enhanced. So, in order to recover the background, the original image $f(x,y)$ is added with the difference, and we get an edge enhanced image with the background features preserved.

Unsharp Masking → Dark-room Photography

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

- $f_s(x,y)$ → Sharp Image
- $f(x,y)$ → Original Image
- $\bar{f}(x,y)$ → Blurred Image / Mean Filtered Image / Rich with Low Frequency Component

High-boost filtering →

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y) \qquad A \geq 1$$

$Af(x,y)$ → Scaled Version of the original image

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y)$$

or, $f_{hb}(x,y) = Af(x,y) + f_s(x,y) - f(x,y)$

or, $\boxed{f_{hb}(x,y) = (A-1) f(x,y) + f_s(x,y)}$

Gradient: $\nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j}$

Laplacian: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2}\hat{i} + \frac{\partial^2 f}{\partial y^2}\hat{j}$

Gradient is a Vector

Laplacian is a Scalar

This produces a vector which has two components g_x and g_y which represents the maximum change in two directions x, y.

Sobel Operators:-

Let say the image values are,

z1	z2	z3
z4	z5	z6
z7	z8	z9

Now, the kernels can be like,

-1	0	1

or,

-1
0
1

$g_x = z_3 - z_1$

$g_y = z_7 - z_1$

This will keep the pixel z2 at the center and produce the edge at that (x,y) pixel having $v(x,y) = z_2$.

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$ More Weight on the center pixel $$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$ SOBEL OPERATORS

$mask_x$ $mask_y$

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$ No Weight on the center pixels $$\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$ PREWITT OPERATORS

$mask_x$ $mask_y$

This is used to produce the edge in the x direction

This is used to produce the edge in the y direction

Now, we can define kernels with non-equal weights. As we move fate away from the centre pixel, the weight keeps on reducing.

$-1/4$	$-1/3$	0	$+1/3$	$1/4$
$-1/3$	$-1/2$	0	$1/2$	$1/3$
$-1/2$	-1	0	1	$1/2$
$-1/3$	$-1/2$	0	$1/2$	$1/3$
$-1/4$	$-1/3$	0	$1/3$	$1/4$

This is an example of non-equal weight mask$_x$, which produces the edge in the ← direction (x direction)

Gradient $\nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j}$ applied on a function, produces an edge, and the strength of the edge is found by $|\nabla f|$

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

For computational ease, we can also perform,

$$|\nabla f| \simeq \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$$ Computionally cheaper but not accurate

Now, Angle of the edge $= \tan^{-1}\left(\frac{g_x}{g_x}\right)$.

For finding out, diagonal edges, we can define the kernels like,

$$\text{mask}_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{mask}_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$ ROBERT OPERATOR

It produces two diagonal edges in two direction

P.T.O.

Image Enhancement of an Image with challenges,

(i) Dynamic Range of Image is Narrow
(ii) The Image is Noise Predominant

Original Image P

Step 1: Perform **Laplacian**, thin edges are emphasised, but noise is enhanced too, **sum it up** with the original
Step 2: Perform **Gradient** Operator, thick edges are emphasised, there are noises too
Step 3: Perform the **Mean Filtering** Operation on the Gradient Image and clear the noise.
Step 4: **Multiply** the smoothened image with the summed image in Step 1
Step 5: **Add** the Step 4 image with the original image
Step 6: Perform **Power Law Transformation** to improve the dynamic range

Spatial Image Processing →

(i) Point Processing: $S = T(r)$ where, $r = f(x, y)$ Input Intensity
$s = g(x, y)$ Output "
T = operator defined on f

Eg: Linear Stretching

(ii) Neighborhood Processing: $g(x,y) = T[f(x,y)]$

where, $f(x,y)$ = Input Image
$g(x,y)$ = Output Image
T is an operator defined on f over the neighborhood of the pixel (x,y)

Eg: Mean Filtering

Functions used in Spatial Domain Image Processing:

(i) Linear function $S = L - 1 - r$

(ii) Logarithmic $S = c\log(1+r)$

(iii) Power-law (n^{th} power, n^{th} root) $S = cr^{\gamma}$

Bit Plane Slicing →

- Lower Order Bit Plane → Contribute to subtle intensity details
- Higher Order Bit Plane → " " visually significant data

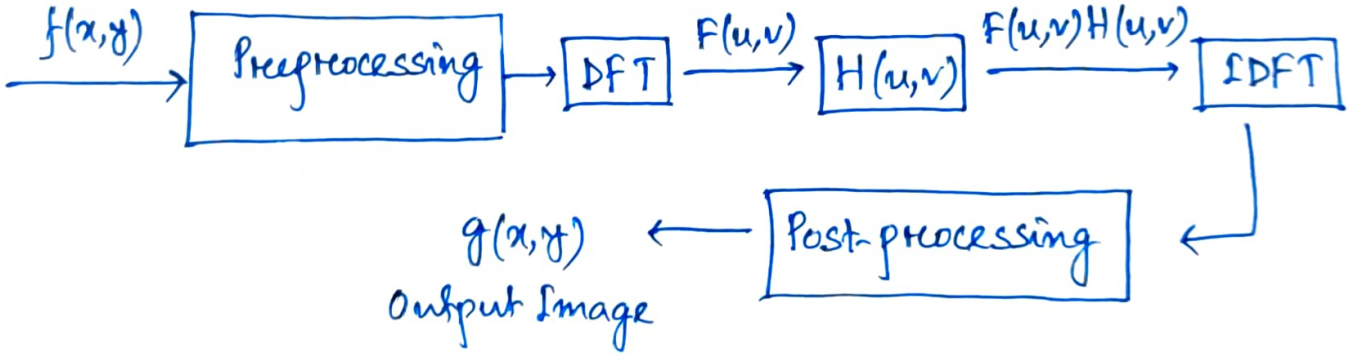
21st March, 2022

Frequency Domain Image Enhancement :-

Input Image $f(x,y)$; Output Image $g(x,y)$

Basic Steps :

Preprocessing
1. Multiply Input Image $f(x,y)$ by $(-1)^{x+y}$
   $\Rightarrow$ to center the frequency at $u = \frac{M}{2}$, $v = \frac{N}{2}$
2. Perform Discrete Fourier Transform on $(-1)^{x+y} f(x,y)$
3. Apply Filter $H(u,v)$ on Step 2
4. Take real part of Step 3
5. Perform Inverse DFT
6. Multiply results of Step 5 by $(-1)^{x+y}$ (Post-Processing)



Basic Steps of Freq. Domain Img. Enh.

As we know, $f(x,y) = i(x,y) \cdot r(x,y)$

$\Rightarrow F[f(x,y)] \neq F[i(x,y)] F[r(x,y)]$

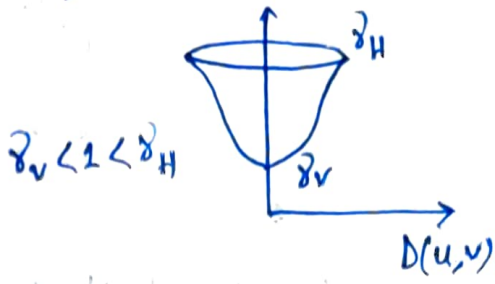
① So, in order to separate illumination (low freq component) and reflectance (high freq component), we first pre process by applying natural logarithm.

$$\log_e f(x,y) = \log_e i(x,y) + \log_e r(x,y).$$

By this way, we are meeting two objectives,

(i) Dynamic Range Compression

(ii) Contrast Enhancement



② Now, we apply DFT,

$$F[\log_e f(x,y)] = F[\ln i(x,y) + \ln r(x,y)]$$

$$\text{or, } Z(u,v) = F[\ln i(x,y)] + F[\ln r(x,y)]$$

$$= I(u,v) + R(u,v)$$

③ Now, we apply the filter $H(u,v) \Rightarrow$

$$Z(u,v) \cdot H(u,v) = I(u,v) \cdot H(u,v) + R(u,v) H(u,v)$$

④ Then, we perform IDFT $\Rightarrow$

$$F^{-1}[S(u,v)] = F^{-1}[I'(u,v) + R'(u,v)]$$

$$\text{or, } S(x,y) = F^{-1}[I'(u,v)] + F^{-1}[R'(u,v)]$$

$$= i'(x,y) + r'(x,y)$$

⑤ Finally, we do post-processing,

$$g(x,y) = e^{S(x,y)} = e^{i'(x,y)} \cdot e^{r'(x,y)} = i_0(x,y) \cdot r_0(x,y)$$

24th March, 2022

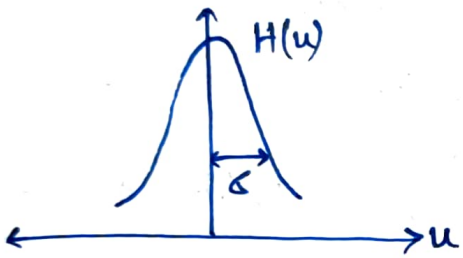
Gaussian Function → Why it is so important?

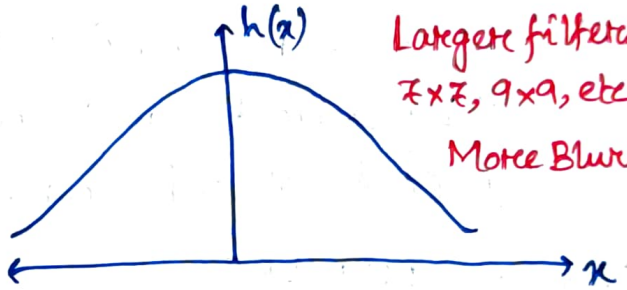
$$H(u) = Ae^{-u^2/2\sigma^2}$$

$$h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 x^2}$$

i) Both the functions $H(u)$ and $h(x)$ are Gaussian in nature and are real functions

ii)





Larger filters
7×7, 9×9, etc
More Blurring

Reciprocal effect means if $H(u)$ is narrow in width, $h(x)$ is broader

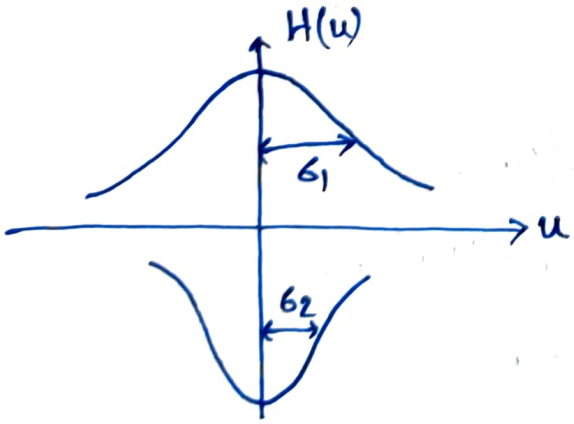
Both are positive, so Smoothening Operation

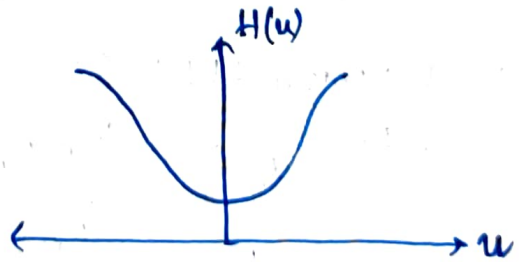
Low Pass to High Pass →

$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2}$$

These conditions dictate the nature of $H(u)$, $h(x)$

$$A \geq B, \quad \sigma_1^2 > \sigma_2$$

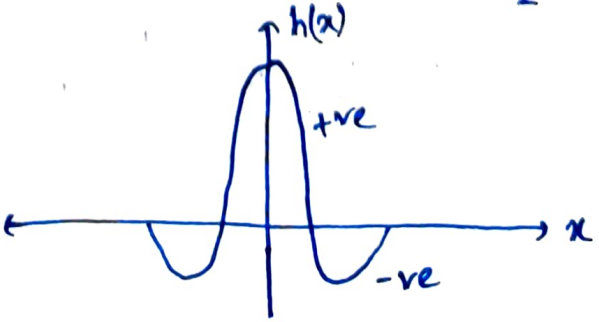




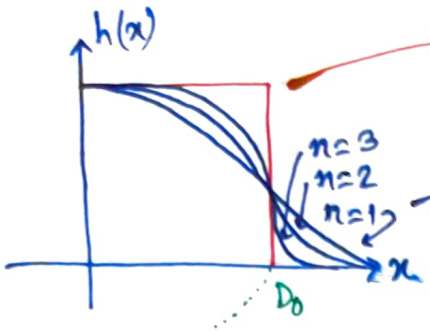
In Spatial Domain,

$$h(x) = \sqrt{2\pi}\sigma_1 A e^{-2\pi^2\sigma_1^2 x^2} - \sqrt{2\pi}\sigma_2 B e^{-2\pi^2\sigma_2^2 x^2}$$

Both Positive and Negative,
So Sharpening Operation



Low Pass Filters



(i) Ideal Filter
(ii) Butterworth Filter
(iii) Gaussian Filter

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

⇒ Ideal Low Pass Filter in 2D

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

⇒ Gaussian Filter

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

⇒ Butterworth Low Pass Filter

n = Filter Order
D_0 = Cutoff Frequency

For High Pass filters →

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

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

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

The design criteria D_0 is determined by the α-factor ie the fraction of power of the image to be retained after filtering.

(i) $\mathcal{F}[f(x)] = F(u)$

(ii) $\mathcal{F}\left[\frac{df(x)}{dx}\right] = juF(u)$

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

For Laplacian, $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

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

Taking Fourier Transform on both sides

$$= [(ju)^2 + (jv)^2] F(u,v)$$

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

$$\therefore \boxed{H(u,v) = -(u^2+v^2)}$$ Laplacian in Frequency Domain

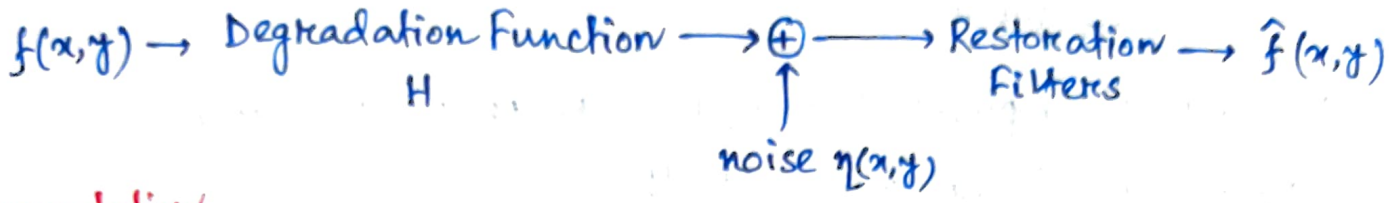
$H(u,v) = -\left[(u-\frac{M}{2})^2 + (v-\frac{N}{2})^2\right]$, centered at $(\frac{M}{2}, \frac{N}{2})$

Unsharp Masking: $f_{hp}(x,y) = f(x,y) - f_{lp}(x,y)$

Highboost Filtering: $f_{hb}(x,y) = Af(x,y) - f_{lp}(x,y)$

$$\boxed{H_{hp}(u,v) = (A-1) + H_{hp}(u,v)}$$ → Proof?

IMAGE RESTORATION



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

The restoration process is a Blind Deconvolution as we do not have any knowledge about $h(x,y)$

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

If $H(u,v) = 1$, $G(u,v) = F(u,v) + N(u,v)$ No Degradation

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

Noise Models → Gaussian, Rayleigh, Uniform

Sinusoidal Noise → Pattern in Freq. Domain

In order to clear we will require, Band Reject Filter

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) < D_0 \text{ and } D_2(u,v) < D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$\hat{f}(x,y) = F^{-1}[G(u,v)H(u,v)]$$

There could be possibility that there are multiple frequency components of the sinusoidal noise.

Notch Reject Filters → Used to suppress some frequency component

Notch Reject Filter → The $\eta(x,y)$ is an interference pattern having fundamental frequency and its harmonics.

If $\eta(x,y)$ is completely known, we can apply subtraction

But, $\eta(x,y)$ is not known at all.

So, in the frequency domain, we use the Notch Pass Filter

$F^{-1}[H(u,v)\,G(u,v)] = \hat{\eta}(x,y)$ Estimated Noise (fundamental Freq Component)

$$\therefore \hat{f}(x,y) = g(x,y) - w(x,y)\,\eta(x,y)$$

$w(x,y)$ ↓ Weight of the Components

Optimum Notch Reject Filter →

Now, we define a sub-image of $(2a+1), (2b+1)$

$$\sigma^2_{Sub}(x,y) = \frac{\sum_{s=-a}^{+a}\sum_{t=-b}^{+b}\left[\hat{f}(x+s,y+t) - \overline{\hat{f}(x,y)}\right]^2}{(2a+1)(2b+1)}$$

Variance

Now, $\overline{\hat{f}}(x,y) = \frac{1}{(2a+1)(2b+1)}\sum_{s=-a}^{a}\sum_{t=-b}^{b}\hat{f}(x+s,y+t)$

Mean

Now, $g(x,y) = \hat{f}(x,y) + w(x,y)\,\eta(x,y)$

or, $\hat{f}(x,y) = g(x,y) - w(x,y)\,\eta(x,y)$

Substituting it,

$$\sigma^2_{Sub}(x,y) = \frac{1}{(2a+1)(2b+1)}\sum_{s=-a}^{+a}\sum_{t=-b}^{+b}\Big[g(x+s,y+t) - w(x+s,y+t)\,\eta(x+s,y+t) - \overline{g(x,y) - w(x,y)\,\eta(x,y)}\Big]^2$$

Here, based on the assumption that within the neighbourhood, pixel values are highly correlated, we write, $w(x+s,y+t) \simeq w(x,y)$

This implies,

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum\sum \left[g(x+s, y+t) - w(x,y)\eta(x+s, y+t) - \overline{g(x,y)} - \overline{w(x,y)}\,\overline{\eta(x,y)} \right]^2$$

Now, our goal is to minimize the variance

$$\therefore \frac{\partial \sigma^2(x,y)}{\partial w} = 0 \Rightarrow w(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \overline{g(x,y)}\,\overline{\eta(x,y)}}{\overline{\eta^2(x,y)} - \{\bar{\eta}(x,y)\}^2}$$

Our goal was to model the optimum weight ↗

Linear Position Invariant Degradation →

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

or, $g(x,y) = H[f(x,y)]$, assuming $\eta(x,y) = 0$

We can think the image as a collection of impulses

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\,\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta$$

Assuming the degradation to be Linear,

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\,h(x-\alpha, y-\beta)\,d\alpha\,d\beta$$

21/03/2022

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

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

↳ Our goal is to know about $h(x,y)$ or $H(u,v)$

Different Approaches for the Estimation of $h(x,y)$

(i) Observation
(ii) Experimentation
(iii) Modeling

By Observation:

We take a sub-image of the degraded image, having a simple structure and strong signal component. We sample pixels from the sub-image $g_s(x,y)$ and generate $\hat{f}_s(x,y)$

From there we can find, $H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$

Then we extend this to the entire image.

By Experimentation:

Get the instrument, responsible for the degradation, and set different settings and choose the one which causes similar degradation. Then pass an impulse of the image through that instrument, and get the degradation.

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

Perform multiple experiments to get a good estimate

By Mathematical Modeling:

Atmospheric Turbulence Degradation = $\boxed{e^{-k(u^2+v^2)^{5/6}}}$

Mathematically modelled in 1964

k = small → Turbulence is mild
k = large → Severe Turbulence

k here is to be estimated

MOTION BLURRING

Relative Motion between the scene and the capturing platform

Planar or Linear Motion causes displacement

$f(x,y)$ at time t, $x - x_0(t)$ during exposure

camera velocity $(x_0(t), y_0(t))$ $y - y_0(t)$, these are the displaced co-ordinates

$$\therefore g(x,y) = \int_0^T f(x - x_0(t),\ y - y_0(t))\, dt$$

T: time between shutter on and off, exposure time

$$G(u,v) = F[g(x,y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_0^T f(x - x_0(t),\ y - y_0(t))\, e^{-2\pi j(ux+vy)}\, dx\, dy\, dt$$

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

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

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

$\rightarrow H(u,v)$ The Degradation

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

for constant motion $(x_0(t), y_0(t)) = (at, bt)$

Inverse filter: $\hat{F}(u,v) = \dfrac{G(u,v)}{H(u,v)} = F(u,v) + \dfrac{N(u,v)}{H(u,v)}$

Noise is enhanced when $H(u,v)$ is small

To avoid the side effect of noise enhancement, we can apply this formulation to freq component (u,v) within a radius D_0 from the center of $H(u,v)$

→ Smaller D_0 → More Blurring

Wiener Filter: Minimum Mean Square Error Filter

Optimize the mean square error $e^2 = E\{(f-\hat{f})^2\}$

Here, the design variable is the $H(u,v)$

Constrained

Least Square Method, ~~Constant~~ Least Square Method:

Minimisation of $C = \sum\sum \nabla^2 f$

Subject to the constraint $\|g - H\hat{f}\|^2 = \|\eta\|^2$

g → Degraded Image; $H\hat{f}$ → Re-degraded Image; η → Noise

Adjusting Gamma Value:

$r = g - H\hat{f}$ → Residue,

We want to adjust γ, so that, $\|r\|^2 = \|\eta\|^2 \pm a$, a → Accuracy factor

1. Specify an initial value of γ
2. Compute $\|r\|^2$
3. Stop if $\|r\|^2 = \|\eta\|^2 \pm a$,

   otherwise return Step2 after increasing γ if $\|r\|^2 < \|\eta\|^2 - a$

   or, " decreasing γ if $\|r\|^2 > \|\eta\|^2 + a$

Use the new value of γ to recompute,

$$\hat{F}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2} G(u,v)$$

Recompute r, and repeat.

Geometric Restoration → The geometry is reflected by the spatial relation between a set of pixel representing a feature

Geometric Operation:

Movement of the pixel from one part to another in a carefully constrained manner.

Step 1: Develop a Transformation function T_x, T_y

$$x' = T_x(x,y), \quad y' = T_y(x,y)$$

Step 2: Implementation of the Function on all pixels

Step 3: Copying the intensity value from old location to new location

Reasons for Geometric Distortion:

① To restore from the distortion during Capturing

② For the registration of image — Deliberate Distortion

Affine Transformation:

$$T_x = a_0x + a_1y + a_2$$
$$T_y = b_0x + b_1y + b_2$$

⇒ Straight Line remains Straight line
Parallel Lines remain Parallel lines

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Representation of the General Framework

Example:	a_0	a_1	a_2	b_0	b_1	b_2
Translation by $\Delta x, \Delta y$	1	0	Δx	0	1	Δy
Scaling	S	0	0	0	S	0
Rotation	$\cos\theta$	$-\sin\theta$	0	$\sin\theta$	$\cos\theta$	0

For Implementation of any Arbitrary Transformation :-

We take 3 landmark points (6 variable, so 6 equations)
We find their position in the output image
Develop the equation and solve to get transformation coefficients
Apply the transformation to the entire image

Forward Mapping Approach : Rotation

Create an o/p image g, of size M×N ~~for all pixels (x,y). do~~

```
for all pixels (x,y) do
        g(x,y) = 0
end for
a0 = cosθ, a1 = -sinθ, b0 = -a1, b1 = a0
for all (x,y) in f do
        x' = round(a0x + a1y)
        y' = round(b0x + b1y)
if (x',y') lies within g
        g(x',y') = f(x,y)
end if
end for
```

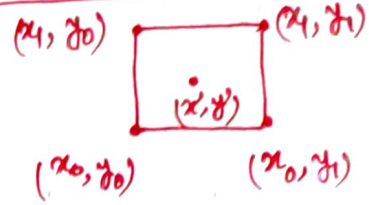
Backward Mapping Approach: Finding corresponding (x,y) from (x',y')

1. Nearest Neighbor or Zero-Order Interpolation
2. Bilinear Interpolation or First " "
3. Higher Order " or Cubic Interpolation

Location

Zero-Order → Round-off the new pixel value after applying the transformation function, and copy the intensity value.

First Order →



$$f(x',y') \simeq f(x_0,y_0) + [f(x_0,y_1) - f(x_0,y_0)]\Delta y + [f(x_1,y_1) - f(x_0,y_0)]\Delta x \Delta y + [f(x_1,y_0) - f(x_0,y_0)]\Delta x$$

$\Delta x \simeq x' - x_0$

$\Delta y \simeq y' - y_0$

$P \simeq [f(x_0,y_1) - f(x_0,y_0)]\Delta x \quad \simeq P + (Q-P)\Delta y$

$Q \simeq [f(x_1,y_1) - f(x_0,y_0)]\Delta x$

Higher Order → Cubic Transformation, 16x16 neighbour

Large Neighbourhood Averaging Operation → Blurring Noise

So, Nearest Neighbour gives Averaging Operation

Farther " " Difference Operation

LOG → Laplacian of Gaussian → Averaging Operation
→ Differentiation

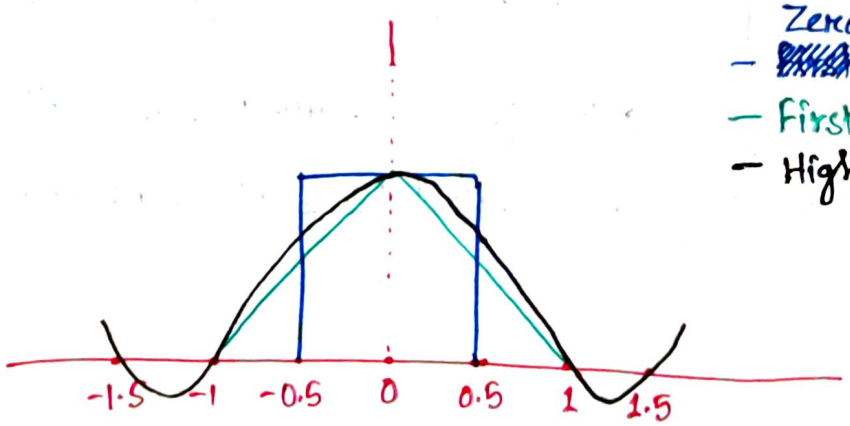


IMAGE SEGMENTATION

Partitioning the entire image into different regions.

Set of pixels
having resemeblance
wrt some spatial relation

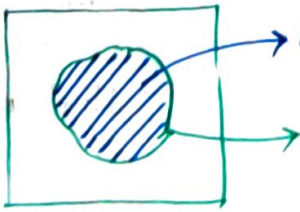
Two issues → (i) Which attribute to be thresholded?
(ii) What is the optimal value of the threshold?

Non-contexual Thresholding → Gives label to each pixel of one class irrespective of the connectivity simply based on the attribute value.

Contexual Thresholding → Gives importance to two aspects:
(i) Attribute similarity (ii) Connectivity

4-Neighbors → For a set of pixels $\{p_1, p_2, \ldots, p_n\}$, if any two pixel p_i, p_{i+1} are 4-neighbors to each other then the set of pixels are 4-neighbors to each other.

Contexual Thresholding → Works on
(i) Concept of Similarity (ii) Concept of Discontinuity



→ Concept of Similarity: Pixels with uniformity
→ Concept of Discontinuity: Boundary b/w regions

Region Similarity → Let, R be a region with pixels (i,j), (m,k), and P is a uniformity predicate, defined as,

$$P(R) = \begin{cases} \text{if } |f(i,j) - f(m,k)| \leq \Delta, \text{ then true} \\ \text{false, otherwise} \end{cases}$$

Two things are important, connectivity b/w (i,j), (m,k), and Δ value

This can be modified as, $P(R) = \begin{cases} \text{True, if } |f(i,j) - \mu_R| \leq \Delta \\ \text{false, otherwise} \end{cases}$

μ_R is the mean value of all the intensity values in R except (i,j)

Region Growing → We start with some initial seed pixels. A pixel to be included in a region,

(i) if the pixel has not been ~~already~~ assigned to other region
(ii) if the pixel has some form of connectivity with the region
(iii) if the inclusion of the pixel does not violate the uniformity of the region.

Let $f(x,y)$ be an image for which regions are to be grown
Define $R_1, R_2, R_3, \ldots, R_m$ of regions, each region contains one seed pixel.

```
for all Ri do
  for all pixel p at the borders of Ri do
    for all neighbors of p do
      Let (x,y) be the neighbor's coordinate
      if f(x,y) is not assigned and |f(x,y) - μi| ≤ Δ
          update μi, include (x,y) in Ri
      end if
    end for
  end for
end for
repeat until no more pixels are being assigned to regions
```

Complete Segmentation must satisfy the criteria :-

* (i) All pixels must be assigned to some region
* (ii) Each " " " " " a single region
(iii) Region must be a set of connected pixels
(iv) Inclusion of a pixel should not violate the region similarity
* (v) Merging two region must be non-uniform

Region Growing Technique does not ensure (i), (ii), (v).

Split and Merge Method →

Image Compression

A form of compact representation of data

Provided there are some redundancy

(i) Lossless Compression → Throw out the inherently present redundancy. When we try to decompress, the original image is retrieved completely without loss of any information.

Issues associated with Compression:

(i) Storage, (ii) Transmission time/Bandwidth (Delay)
(iii) Power

(ii) Lossy Compression → Here, we look for a representation space and create redundancy and exploit it.

But when we try to decompress, even though we do not get the exact original image, the loss is not objectionable in human eye.

Types of Redundancy:

(i) Coding Redundancy:

(ii) Interpixel Redundancy:

Pixel values are highly correlated, i.e, one pixel value can be predicted from others.

(iii) Psychovisual Redundancy:

The resolution of our eye is also finite. Human eye cannot distinguish close pixel values like 200, 201, 202 from each other.

(iv) Interframe Redundancy: Applicable for Video-like signal

Analogous to Interpixel Redundancy in still image.

Performance Assessment of Image Compression :-

$$\text{Compression Ratio} = \frac{\#\text{bits to represent the compressed dataset}}{\#\text{bits to represent the original dataset}}$$

(i) ↓ How much compactness can be achieved?

(ii) Time required for compression and decompression.
Sometimes, compression time should be less.
" decompression " " " "

(iii) Lossy Compression : Distance b/w original, decompressed image

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

Root Mean Square Error ⇒ Objective Assessment
↓
Not necessarily tallies with Subjective Assessment

Lossy Compression Techniques →