IMAGE ENHANCEMENT

The objective of image enhancement is to improve the quality of image. Enhancement techniques enhance the quality of image to improve visual perception.

Enhancement techniques are broadly classified into two categories.

(1) Spatial Domain operations - These operations work directly on pixel itself in an image & manipulates them. Spatial domain operations are broadly classified into two categories

a) Point Processing or Point operations
b) Mask processing or Mask operations

(2) Frequency Domain Operations:-

The fourier transforms are applied on image inorder to modify the coefficients. Then inverse fourier transforms are applied.

Consider a pixel in an image. $f(x)$.

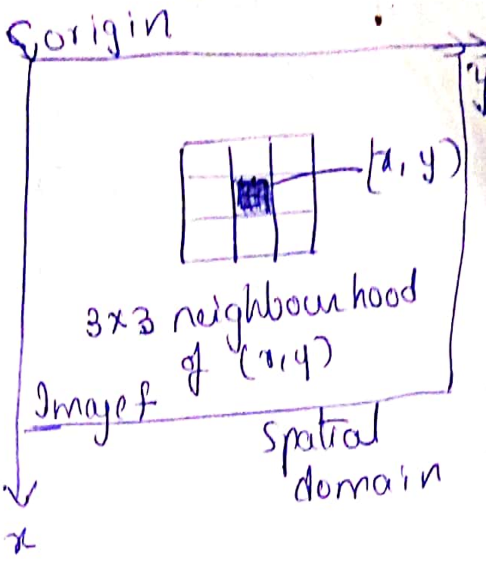
$g(x) = T[f(x)]$ - onedimensional.

for $f(x,y) \rightarrow g(x,y) = T[f(x,y)]$ - two dimensional.

T - transformation

$g(x)$, $g(x,y)$ — represents processed o/p image.

Gray Level Transformation (Intensity Mapping):



Let us consider a rectangular image. Take ~~any~~ a ~~one~~ pixel at location (x,y) in the image. The neighbour hood of this pixel is a subimage around point (x,y) in image f.

The size of the neighbour hood is 3x3. It may be 1x1, 3x3, 5x5, 7x7 etc.

In case of point processing, size of neighbour hood is 1x1

Therefore transformation $s = T(r)$

SPATIAL DOMAIN TECHNIQUES

1) Image Negative
2) Contrast stretching
3) Log Transformation
4) Level Slicing
5) Bit plane slicing
6) Power Law transformations
7) Histogram specification
8) Histogram Equalization

→ Point Processing Techniques

(i) Image Negative :-

Sometimes, based on illumination sources, images may be too bright or too dark. Hence it becomes too difficult to extract information from image. Hence Image negative is considered. This transformation reverses light & dark conditions.

The no. of intensity or gray levels in an image are represented by 'L' starting from $0 \rightarrow L-1$ varying in steps of 1.

we know intensity levels $L = 2^b$ where b- bitwidth

$\therefore$

$$S = T(r) = L - 1 - r$$

when $r = 0$; $S = L - 1$ max. intensity

$r = L-1$; $S = 0$

where r = Intensity levels on x - axis.

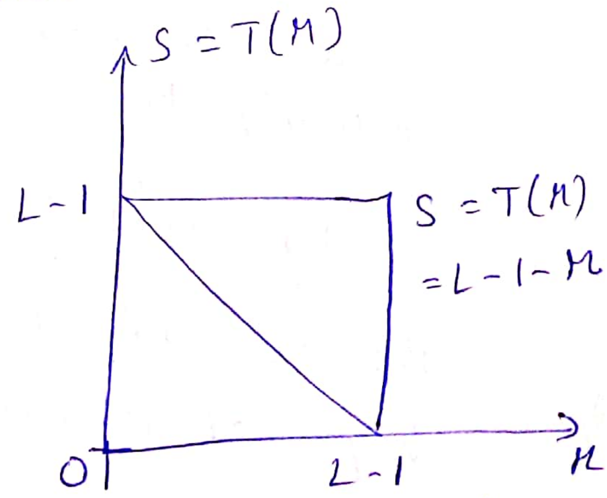
Ex:- If $b = 8$; $r = 0$;

$S = T(r) = 2^8 - 1 - 0$

$\Rightarrow S = 255$

$b = 8$; $r = L - 1$

$S = T(r) = 2^8 - 1 - 2^8 + 1$



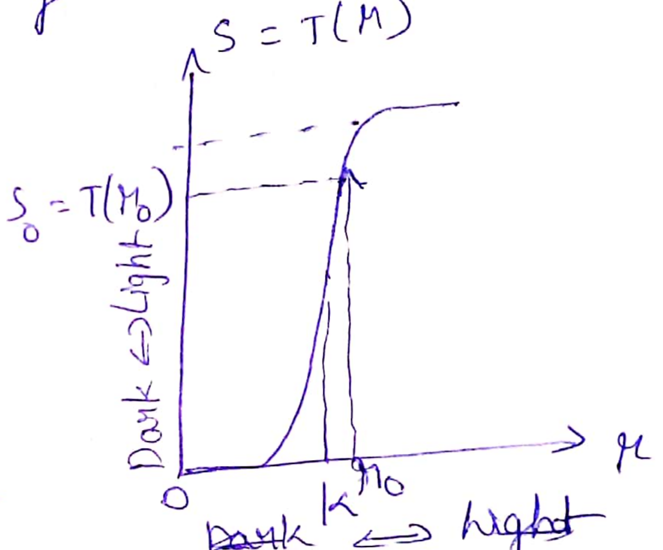
Hence maximum intensity value can be converted to minimum & min. intensity value to maximum.

the Transformation is a straight line with a slope of -45° passing through $(0, L-1)$ & $(L-1, 0)$

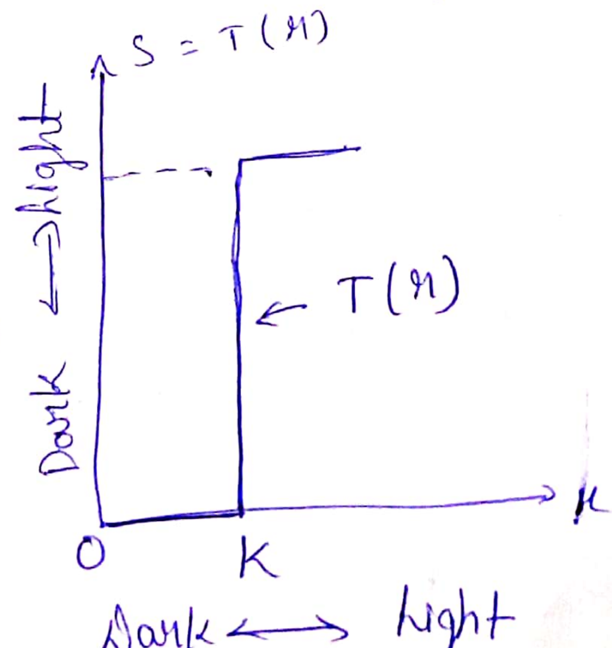
Enhancement techniques applied to one image may not suit for another image.

(2) Contrast Stretching :-

the process of expanding the range of intensity levels in an image is known as contrast stretching.



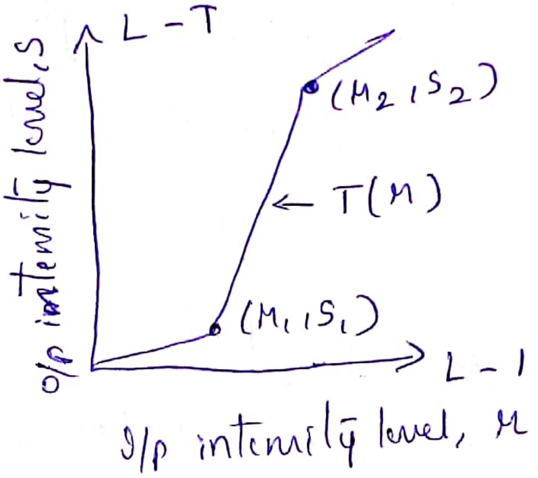
In fig. x-axis represents intensity values 'r' of image & y-axis represents intensity values of processed image. Here we can see that above & below threshold there is a variation in intensity levels.



By applying transformation the values below k are compressed towards black & above k are compressed towards white

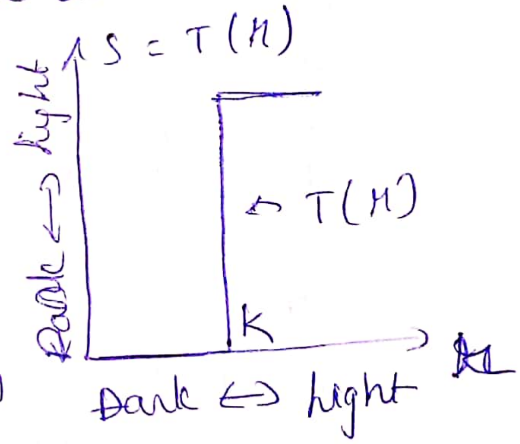
For ex let us consider a low contrast image & transformation used for contrast stretching

Case 1:- If $r_1 = s_1$; $r_2 = s_2$ transformation function is linear & produces no changes in intensity levels



Case 2:- If $r_1 = r_2$;

$s_1 = 0$; $s_2 = L-1$ transformation becomes a thresholding function which creates a binary image.

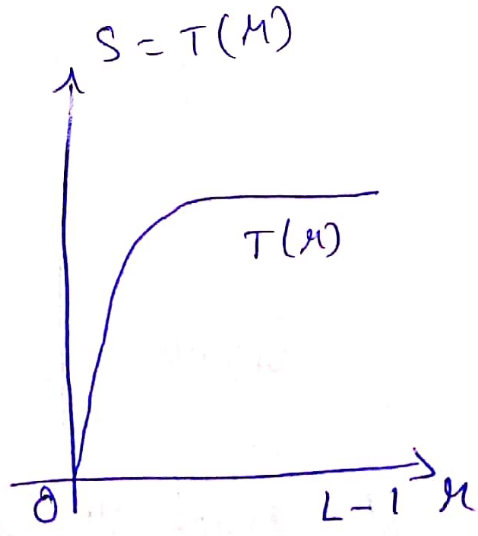


$\therefore$ $(r_1, s_1) = (r_{min}, 0)$

$(r_2, s_2) = (r_{max}, L-1)$

Thus transformation stretches image to $(0, L-1)$

(3) Log transformations:-



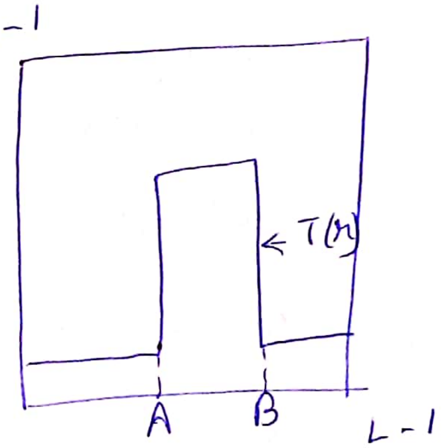
The general form of log transformation is expressed as

$$s = T(r)$$

$$\Rightarrow \boxed{s = c \log(1+r)}$$

where c = const; $r \geq 0$

these transformations map a narrow range of low gray level values in the input image into a wider range of o/p levels. They compress the dynamic range of images with large variations in pixel values



(4) Level Slicing :-

Level slicing highlights the specific range of gray values.

There are two ways for level slicing

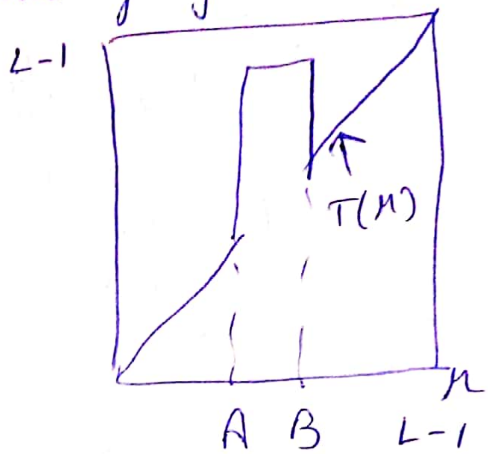
(a) Level slicing without preserving the background :-

Here the level slicing is done without preserving the background details. The gray levels of a particular range are displayed with high values and others with low values. Hence, in figure only one part is highlighted as zeroes

$$S = T(r) = L \; ; \quad A \leq r \leq B$$
$$= 0 \quad \text{otherwise}$$

Drawback :- In this method background information is discarded.

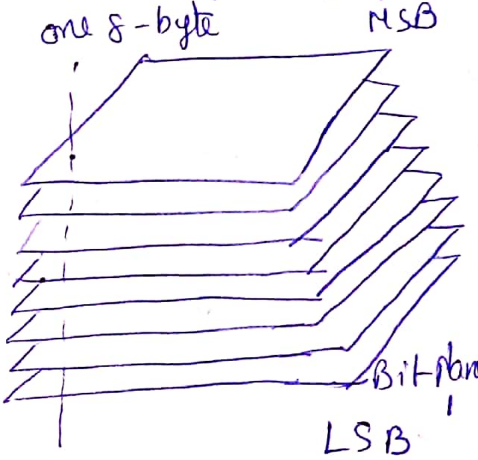
(b) Level slicing with background :- This method preserves background details. High values are displayed for a particular range & original gray values in other areas



$S = T(r) = L \; ; \; a \leq r \leq b$

$= r \; ; \text{ otherwise}$

(5) Bit Plane Slicing :- In this method, images are divided according to no. of bits. The lower planes ~~bits~~ are considered as bit plane 0 (LSB bits) & higher planes are considered as bit plane 7 (USB bits)



Here take a gray level image & convert it into binary image. Represent the image with fewer bits & compress it to small size. Then enhance the image by focusing

For 8 bit image 0 is encoded as 00000000 &
1 is encoded as 11111111

Any no. b/w 0 & 255 is encoded as one byte

Advantage :- Image can be compressed.

b) Power Law Transformation :-

Power transformation is used in image capturing devices, printers etc. the transformation is given

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

c, r are constants

$s = (r + \epsilon)^{\gamma}$ for offset purpose (o/p is measured when i/p is zero)

⇒ For γ = s, we have original intensity levels of image at output.

→ For value of γ < 1, transformation function is towards lower intensity side. γ expands dynamic range of small intensity in the i/p image.

→ For value of γ > 1, transformation function is towards higher intensity side. γ

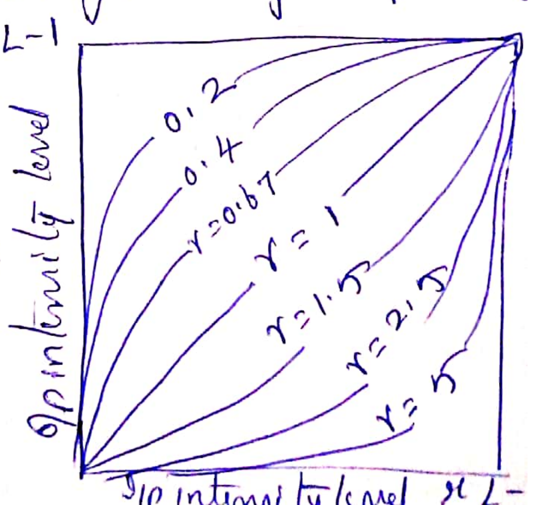
⇒ This correction is called gamma correction

Advantages :-

→ used in variety of devices for image capturing printing & display purpose

→ used for gamma correction

→ Used for general purpose contrast manipulation



(7) Histogram Processing:-

→ Histogram of an image represents the frequency of occurance of gray levels in an image.

→ Histogram manipulation improves quality & provides details about brightness & darkness. (contrast)

→ The values of an image near to origin represent dark ones & those nearer to $L-1$ represent bright ones

→ The histogram of an image with gray levels $[0, L-1]$ is given by

$$\boxed{h(r_k) = n_k} \quad \text{---} \quad (1)$$

where r_k = kth gray level in an image

n_k = No. of pixels having gray level r_k.

→ The normalized histogram is given by

$$\boxed{h(r_k) = \frac{n_k}{n}} \quad \text{---} \quad (2)$$

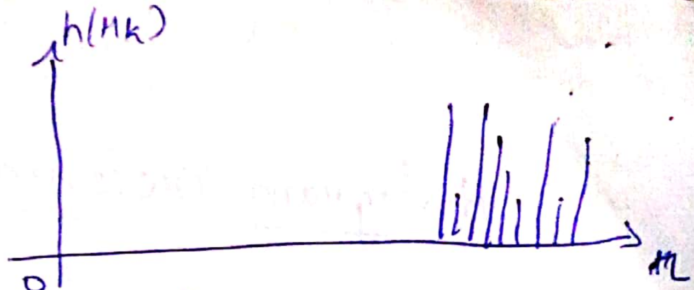
where n - no. of intensity levels.

→ Addition of all terms to normalized histogram is always 1.

→ Applications:- Image enhancement, compression, & segmentation & real time applications

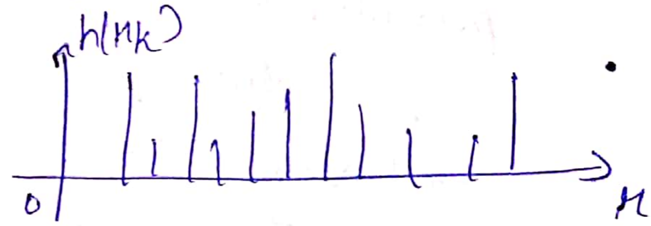
→ A dark image will have a histogram near origin

→ A light image will have histogram very far from origin near max. levels



→ A low contrast image will be narrow & at centre of histogram.

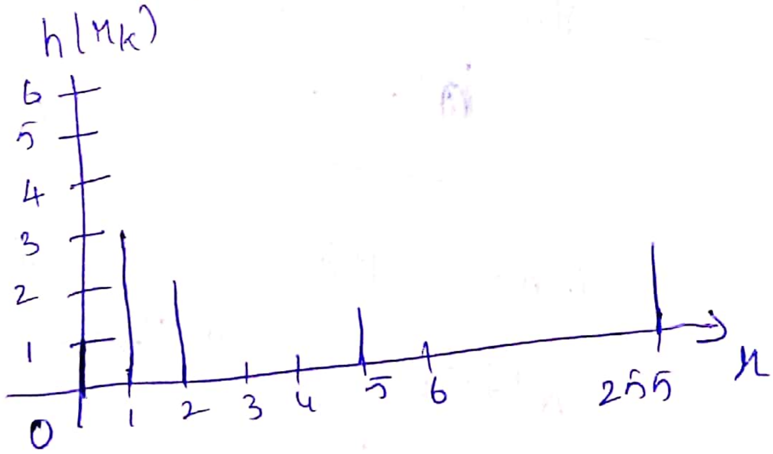
→ A high contrast image will have a wide dynamic range.



→ Histogram calculation is very simple in both hardware & software

Ex :- For a given subimage $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 255 & 255 & 0 \end{bmatrix}$ draw a histogram

In matrix
1 - is repeated thrice
2 - twice
5 - once
255 - twice
0 - once



HISTOGRAM EQUALIZATION :-

Consider $S = T(r)$; $0 \leq r \leq 1$ — (1)

r - original image
T - Transformation
T(r) - single valued monotonically increasing function
If $r = 0$ indicates BLACK
$r = 1$ indicates WHITE
Hence r limit is from $[0,1]$

Since gray levels are randomly distributed in image, we use probability density function.
Let $P(r)$ & $P(s)$ denote probability density function of r & s.

$$P(s)\,ds = P(r)\,dr$$

For $s = T(r)$; $P(s) = P(r)\frac{dr}{ds}$ — (2)

From (1) transform must be such that pixel distribution is uniform.
To obtain uniform distribution we use cumulative distribution function

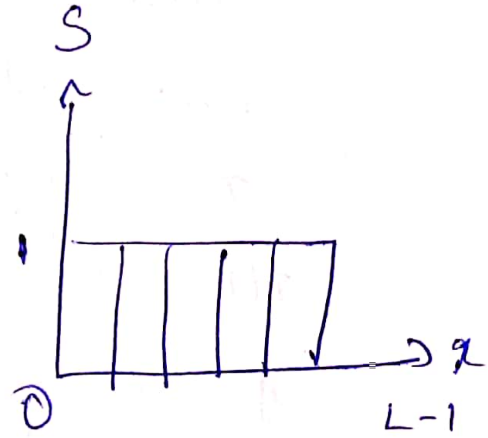
$$P(s) = 0 ; \quad 0 \leq s \leq 1$$

$$s = T(r) = \int_0^r P(w)\,dw$$

where dw - dummy variable

Differentiating

$$\frac{ds}{dr} = \frac{d}{dr}\left[\int_0^r P(w)\,dw\right]$$



$$\Rightarrow \frac{ds}{dr} = P(r) \quad -(3)$$

Substituting in eq (2) ~~(3)~~

$$P(s) = \frac{P(r)}{P(r)} = 1 \quad -(4)$$

The eq. represents uniform equalization

Drawback :-

Accuracy is poor

Histogram Specification or Histogram matching :-

Suppose k represents gray level at someplace in image

$$k = T_1(r) = \int_0^r P(r)\,dr$$

$$k = T_2(r) = \int_0^r P(s)\,ds$$

Hence transformation achieves desired result when

$$s = T_2^{-1}[T_1(r)]$$

Generating histogram of our requirement is called histogram matching

Example:- Perform histogram equalization for the image

$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 4 \\ 3 & 3 & 4 & 5 \\ 4 & 5 & 2 & 5 \end{bmatrix}_{4\times 4}$$

Max. value of pixels in image is 5

No. of bits required to represent 5 is 3 bits (101)

∴ No. of possible gray levels = 8 i.e 0 - 7.

The histogram equalization process is as follows:

Steps to be followed	~~0~~	~~1~~	~~2~~	~~3~~	~~4~~			
1) Gray levels	0	1	2	3	4	5	6	7
2) No. of pixels in each gray level	0	0	1	3	8	4	0	0
3) Cumulative sum	0	0	1	4	12	16	16	16
4) Divide cumulative sum by total no. of pixels	$\frac{0}{16}$	$\frac{0}{16}$	$\frac{1}{16}$	$\frac{4}{16}$	$\frac{12}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$
5) Multiply the result in (4) by total highest value of pixels	0×7	0×7	$\frac{1}{16}\times7$	$\frac{4}{16}\times7$	$\frac{12}{16}\times7$	1×7	1×7	1×7
6) Result rounded off to nearer value	0	0	1	2	5	7	7	7
7) Equivalent values for matrix			$2 \downarrow$	$3 \downarrow 2$	$4 \downarrow 5$	$5 \downarrow 7$		

$\therefore$

I/P image $\begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 4 \\ 3 & 3 & 4 & 5 \\ 4 & 5 & 2 & 5 \end{bmatrix} \xrightarrow{\text{Histogram equalization}} \begin{bmatrix} 5 & 5 & 5 & 5 \\ 7 & 5 & 2 & 5 \\ 2 & 2 & 5 & 7 \\ 5 & 7 & 1 & 7 \end{bmatrix}$

IMAGE ENHANCEMENT BY NEIGHBOURHOOD PROCESSING:

→ Neighbourhood operations work with the values of image pixels in the neighbourhood and corresponding values of a subimage that has same dimensions as the neighbourhood. A neighbourhood is a small rectangle.

→ The subimage is called a filter, mask, kernel template or window.

→ The values in the filter are called as coefficients rather than pixels.

→ The process of spatial filtering consists simply of moving the filter mask from point to point in an image.

→ At each point, the response of the filter is given by sum of products of filter coefficients & the corresponding image pixels in the area spanned by the filter mask.

→ consider a 3x3 neighbourhood. Let centre pixel be represented by $g(x,y)$. The neighbourhood may be of any size i.e 5x5, 7x7 etc.

→ we change the values of $g(x,y)$ based on the values of 8 neighbours i.e $f(x,y) = T[g(x,y)]$.

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

→ 3x3 neighbourhood.

→ fig below shows a mask (or template or window). place this mask on the above image & multiply each component of the mask with corresponding value of the image. Add the values & place the result in the centre.

→ Size of mask & image must be same.

→ If 'g' is the original image, the modified image 'f' is given as

$$f(x,y) = g(x-1,y-1)\,w_1 + g(x-1,y)\,w_2 + g(x-1,y+1)\,w_3 + g(x,y-1)\,w_4 +$$
$$g(x,y)\,w_5 + g(x,y+1)\,w_6 + g(x+1,y+1)\,w_7 + g(x+1,y)\,w_8 +$$
$$g(x+1,y+1)\,w_9.$$

→ once $f(x,y)$ is calculated, shift the mask by one step towards the right to the next pixels.

→ The important operations that can be acheived using neighbourhood processing is Image filtering.

SPATIAL FILTERING TECHNIQUES.

Spatial filtering techniques are broadly classified into two categories

Spatial filters

- Smoothing spatial filters (blurring, noise reduction)
  - Linear filters (operations performed on image pixel)
    1) Averaging filter on Low pass filtering
    2) weighted averaging filter
  - Non-linear or order statistics filter (Based on ranking the pixels).
    1) Median filter
    2) Minimum filter
    3) Max. filter
- sharpening spatial filters
  1) I^st order or gradient masking.
  2) II^nd order or Laplacian masking.
  3) Unsharp masking.
  4) High Boost filtering
  5) Sobeal masking.
  6) Robert masking.

SPATIAL SMOOTHING FILTERS

1) Low pass filter (or) Average masking

→ It is a Linear filter

→ In this the value of every pixel in an image is replaced by the average of intensity level in the local neighbourhood.

→ In this masking all intensity level are same.

→ The general form of average masking is

$$g(x,y) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} W_{ij}\, f(x+i,\ y+j).$$

→ 3x3 Low pass spatial mask.

$$= \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3\times 3}.$$

ADVANTAGES

1. Noise gets reduced.
2. Image gets smoothed.

DISADVANTAGES:

1. Average masking leads to ~~blurri~~ blurring of edges, which are desirable features of an image.
2. Image is corrupted by impulse noise then the impulse noise is attenuated & diffused but not removed.

---

2. WEIGHTED AVERAGE MASKING:-

→ It is a linear filter

→ To prevent blurring at the edges, since edges consist of high pass components, we go for weighted average technique.

→ In this technique the pixels nearest to the center are weighted more than the distant pixels.

→ Since the center pixel has more weight, blurring at edges is reduced.

→ The general Expression is

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

→ Weighted average 3×3 Mask is

$$= \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{3\times3}$$

ADVANTAGES

1. Blurring at sharp edges gets reduced.
2. Noise gets reduced.
3. Image gets smoothed.

---

3. MEDIAN FILTER.

→ It is a non-Linear Technique.

→ Median filter provide excellent noise reduction capabilities than linear smoothing filters

→ Median filters are used to reduce salt and pepper noise.

→ A median filter smoothens the image by utilising the mean of neighbourhood.

→ Median filter perform the following task to find each pixel in the processed image.

1. Arrange the pixels in Ascending (or) Descending order.
2. The median of the sorted value is computed is chosen as the pixel value of processed image

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

1, 2, 3, 4, 5, 6, 7, 8, 8

→ Center pixel 4 is replaced with 5.

→ Spatial sharpening techniques ②

1) I order (or) Gradient filter

→ Image differentiation enhances edges and other discontinuities & de-emphasizes areas with slowly varying intensities.

→ By using Gradient masking we find out the vertical and horizontal thick values only.

→ Gradient function $\nabla f = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$

$$f(x,y) = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right]$$

→ Let us consider an image = $\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}_{3\times 3}$

→ differentiation is nothing but difference between previous and present values.

$$\frac{\partial f}{\partial x} = w_7 - w_4 + w_8 - w_5 + w_9 - w_6 + w_4 - w_1 + w_5 - w_2 + w_6 - w_3.$$

$$= w_7 + w_8 + w_9 - w_1 - w_2 - w_3$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{\partial f}{\partial y} = w_3 - w_2 + w_6 - w_5 + w_9 - w_8 + w_2 - w_1 + w_5 - w_4 + w_8 - w_7$$

$$= w_3 + w_6 + w_9 - w_1 - w_4 - w_7.$$

$$\frac{\partial f}{\partial y} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

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

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

2. II order (or) Laplacian filtering (or) High pass mask.

→ By using II order we find thin lines of an image.

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

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

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

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

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

→ 3x3 Mask.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

ADVANTAGES:-

By using Laplacian masking. brightness increases, once brightness increases we can easily identify the edges and boundaries of image.

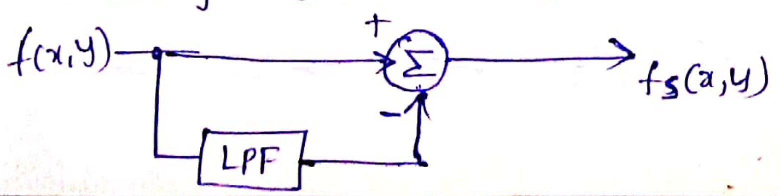
DISADVANTAGE:

Because of Laplacian masking noise gets increased.

3. UNSHARP MASKING:-

→ The main objective of unsharp masking is to increase the contrast of an image.

→ The brightness can be increased by reducing the lowpass components & enhancing highpass components.



$$f_s(x,y) = f(x,y) - f_{LP}(x,y).$$

→ Unsharp masking involves the following steps

1. Blurring the original image.
2. Subtracting the blurred image from original.
3. Add masking to the final image.

---

4. HIGH BOOST FILTERING:

→ For sharpening the image and increase the center pixel value we go for high boost filtering

→ we know that

$$f_s(x,y) = A\,f(x,y) - f_{LP}(x,y)$$
$$= A\,f(x,y) + f(x,y) - f(x,y) - f_{LP}(x,y)$$
$$= (A-1)f(x,y) + f(x,y) - f_{LP}(x,y).$$

$$f_s(x,y) = (A-1)\,f(x,y) + f_{HP}(x,y).$$

→ Examples

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} ; \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(A+4) & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

---

5. HOMO-MORPHIC FILTERING:

$$f(x,y) \rightarrow \boxed{\ln} \rightarrow \boxed{DFT} \rightarrow \boxed{H(u,v)} \rightarrow \boxed{(DFT)^{-1}} \rightarrow \boxed{exp} \rightarrow g(x,y)$$

→ An image $f(x,y)$ can be expressed as the product of its illumination $i(x,y)$ and reflectance $r(x,y)$ components.

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

$$z(x,y) = \ln(f(x,y))$$

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

$$z(x,y) = \ln(i(x,y)) + \ln(r(x,y))$$

$$Z(u,v) = DFT\{\ln i(x,y)\} + DFT\{\ln r(x,y)\}.$$

$$Z(u,v) = F_i(u,v) + F_r(u,v)$$

→ where $F_i(u,v)$ and $F_r(u,v)$ are the fourier transform of $\ln i(x,y)$ and $\ln r(x,y)$.

→ We can filter $Z(u,v)$ using a filter $H(u,v)$

→ So that

$$S(u,v) = H(u,v)\, Z(u,v)$$

$$= H(u,v)\{F_i(u,v) + F_r(u,v)\}.$$

$$S(u,v) = H(u,v)\, F_i(u,v) + H(u,v)\, F_r(u,v).$$

→ The filtered image in the spatial domain is:

$$s(x,y) = IDFT\{S(u,v)\}.$$

$$s(x,y) = IDFT\{H(u,v)\, F_i(u,v)\} + IDFT\{H(u,v)\, F_r(u,v)\}$$

$$i'(x,y) = IDFT\{H(u,v)\, F_i(u,v)\}.$$

$$r'(x,y) = IDFT\{H(u,v)\, F_r(u,v)\}.$$

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

→ Apply Exponential.

$$g(x,y) = e^{s(x,y)}$$

$$= e^{\{i'(x,y) + r'(x,y)\}}.$$

$$= e^{i'(x,y)} \cdot e^{r'(x,y)}.$$

$$\boxed{g(x,y) = i_0(x,y) \cdot r_0(x,y)}$$

→ here

$$i_0(x,y) = e^{i'(x,y)}$$

$$r_0(x,y) = e^{r'(x,y)}.$$

→ By using Homo-morphic filtering we can reduce the effects of the dominant illumination components and reflectance components of the image were sharpened.

6. SOBEAL MASKING:-

→ By using sobeal masking sharp edges can be found it is also similar to gradient filter but the center part is doubled.

$$\frac{\partial f}{\partial x} = w_7 + 2w_8 + w_9 - w_1 - 2w_2 - w_3.$$

$$\frac{\partial f}{\partial y} = w_3 + 2w_6 + w_9 - w_1 - 2w_4 - w_7$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad \frac{\partial f}{\partial y} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

7. ROBERT MASKING:-

→ This masking is also known as "Gradient (or) I order filter"

→ In this masking we take the cross differences.

→ Let us consider an image $= \begin{bmatrix} W_1 & W_2 & W_3 \\ W_4 & W_5 & W_6 \\ W_7 & W_8 & W_9 \end{bmatrix}_{3\times3}$

$$f(x,y) = \sqrt{(w_9 - w_5)^2 + (w_8 - w_6)^2}$$

→ By applying robert masking we can find out the diagonal values

→ Examples for Robert masking.

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

COMBINING SPATIAL ENHANCEMENT METHODS:-

→ We know that to obtain a task. we require applications of several complementary techniques, in order to acheive an acceptable result.

→ The main objective of combining spatial enhancement method is to enhances the image by sharpening it by combining various techniques.

→ we use Laplacian to highlight the prominent edges. & to increase the dynamic range of the intensity level. we use intensity transformation.

→ Median filter is used to reduce salt and pepper noise.

→ The gradient has a stronger response in areas of ramp and step functions.

→ The Laplacian function produces high noise than gradient filter.

→ By using sobeal masking. we can sharp the edges of an image.

→ The smallest possible value of gradient image is '0'.

→ By using the product of Laplacian & smoothed gradient we can increase the sharpness of the image.

→ The type of improvement would not have been possible by using the Laplacian or gradient.

→ The dynamic range can be sharpned by using power low transformation.

→ Level slicing and Bit plane slicing methods are useful for hightlight the part of image.

UNIT-II

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

PRELIMINARY CONCEPTS :- (Basics for fourier transforms)

Signal :- It is a measurable phenomenon that changes over time or throughout space.
Any real signal can be represented in freq. domain

Fourier Series :- Any periodic fn. can be expressed as sum of sines &/or cosines of different frequencies each multiplied by a different coefficients. This sum is called Fourier series.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{j2\pi n t}{T}} \quad \text{— (1)}$$

where T = time period

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-\frac{j2\pi n t}{T}} dt \quad \text{— (2)}$$

$$n = 0, \pm 1, \ldots$$

$$[\, e^{j\theta} = \cos\theta + j\sin\theta \,]$$

Impulses :- A unit impulse of a continuous variable t located at $t=0$, is defined as

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases} \quad \text{— (3)}$$

$$\& \int_{-\infty}^{\infty} \delta(t)\, dt = 1 \quad \text{— (4)}$$

Sifting Property :-

$$\int_{-\infty}^{\infty} f(t)\, \delta t = f(0) \qquad \text{— (5)}$$

If $f(t)$ is continuous at $t = 0$

$$\int_{-\infty}^{\infty} f(t)\, \delta(t - t_0)\, dt = f(t_0) \qquad \text{— (6)}$$

Unit Discrete Impulse :-

$$\delta[x] = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \qquad \text{— (7)}$$

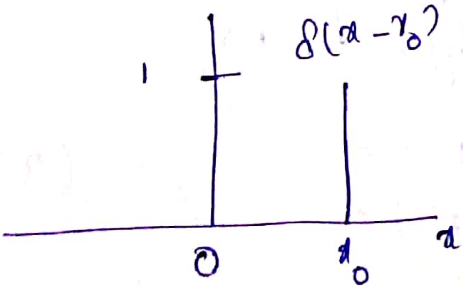
Discrete form :-

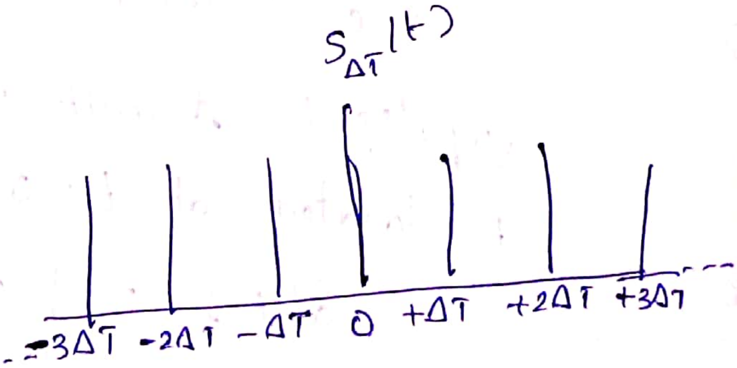
$$\sum_{x=-\infty}^{\infty} f[x]\, \delta[x] = f[x] \qquad \text{— (8)}$$

$$\sum_{x=-\infty}^{\infty} f[x]\, \delta[x - x_0] = f[x_0] \qquad \text{— (9)}$$

Impulse train :- It is defined as the sum of infinitely many periodic impulses ΔT units apart

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T) \qquad \text{— (10)}$$





Convolution :-

Consider two continuous functions f(t) & h(t) of one continuous variable t. The convolution of these two functions is

$$f(t) * h(t) = \int_{-\infty}^{\infty} f(\tau)\, h(t-\tau) \quad -(11)$$

minus sign - for flipping

τ - displacement needed.

$$F[f(t) * h(t)] = H(u)\, F(u) \quad -(12)$$

$$F[f(t)\, h(t)] = H(u) * F(u) \quad -(13)$$

One Dimensional Fourier transform

$$F[f(t)] = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi ut}\, dt \quad -(14)$$

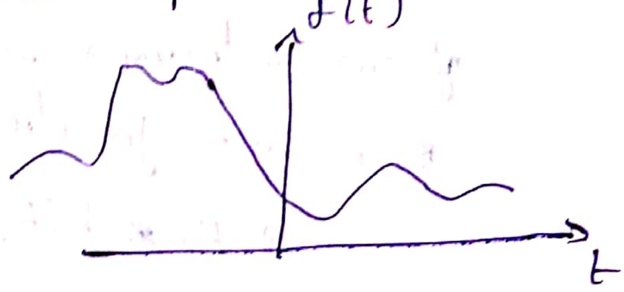
Inverse fourier transform

$$f(t) = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi ut}\, du. \quad -(15)$$

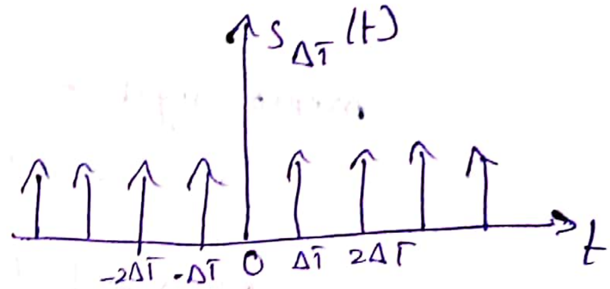
Sampling :-

Consider a continuous function $f(t)$. sample it at uniform intervals (ΔT) of the independent variable t.

Assume function extends from $-\infty$ to ∞ w.r.t 't'.



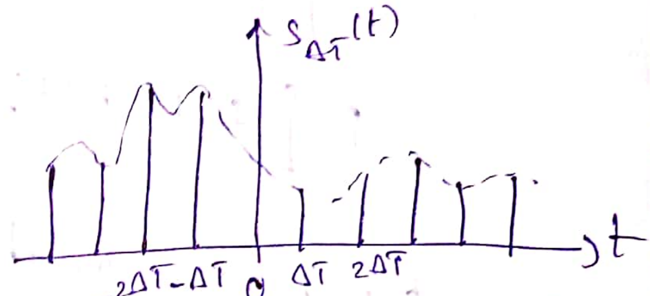
Hence sampling function



$$\tilde{f}(t) = f(t)\, s_{\Delta T}(t)$$

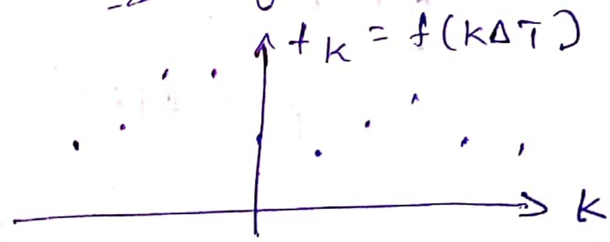
$$= \sum_{n=-\infty}^{\infty} f(t)\, \delta(t - n\Delta T)$$

the strength of the weighted impulse



$$f_k = \int_{-\infty}^{\infty} f(t)\, \delta(t - k\Delta T)\, dt$$

$$= f(k\Delta T)$$



where sifting property of δ is used. & $k = \cdots -2, -1, 0, 1, 2 \cdots$

Fourier transform of sampled function:-

The fourier transform of a sampled function

$$\tilde{F}(u) = F\{\tilde{f}(t)\}$$

$$= F\{f(t)\, s_{\Delta T}(t)\}$$

$$\Rightarrow \tilde{F}(u) = F(u) \star S(u)$$

where $S(u) = \frac{1}{\Delta T} \sum_{n=\infty}^{\infty} \delta\left(u - \frac{n}{\Delta T}\right)$

$$\therefore \tilde{F}(u) = \int_{-\infty}^{\infty} F(\tau)\, S(u-\tau)\, d\tau$$

$$= \frac{1}{\Delta T}\int_{-\infty}^{\infty} F(\tau) \sum_{n=-\infty}^{\infty} \delta\left(u-\tau-\frac{n}{\Delta T}\right) d\tau$$

$$= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} F(\tau)\, \delta\left(u-\tau-\frac{n}{\Delta T}\right) d\tau$$

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

Sampling Theorem & Aliasing.

(from Digital Comm Study)

2) Add orignal image to the image obtained in step 1
the resultant image obtained will be noisy.
the noise in the image has to be reduced.

3) Now noise in the image has to be reduced. one technique to remove noise is using median filter. But median filter is a non-linear filter & hence cannot be used ~~overall~~ because it may remove image features.
An alternate process is to use a mask formed from gradient of orignal image. Apply Laplacian & Gaussian mask to remove noise.

4) Apply sobel masking in order to obtain sharp edges in image.

5) Then smoothen the image using average filter.

6) Multiply the product of Laplacian (step 3 & smoothened image (step 5). Noise will be reduced & we can observe strong edges.

7) Now add the product obtained image in step 6 to the original image. The o/p will be a sharpened image.

~~8)~~ ~~The~~

BASICS OF FILTERING IN FREQUENCY DOMAIN

The two dimensional discrete fourier Transform (DFT) is given by eq

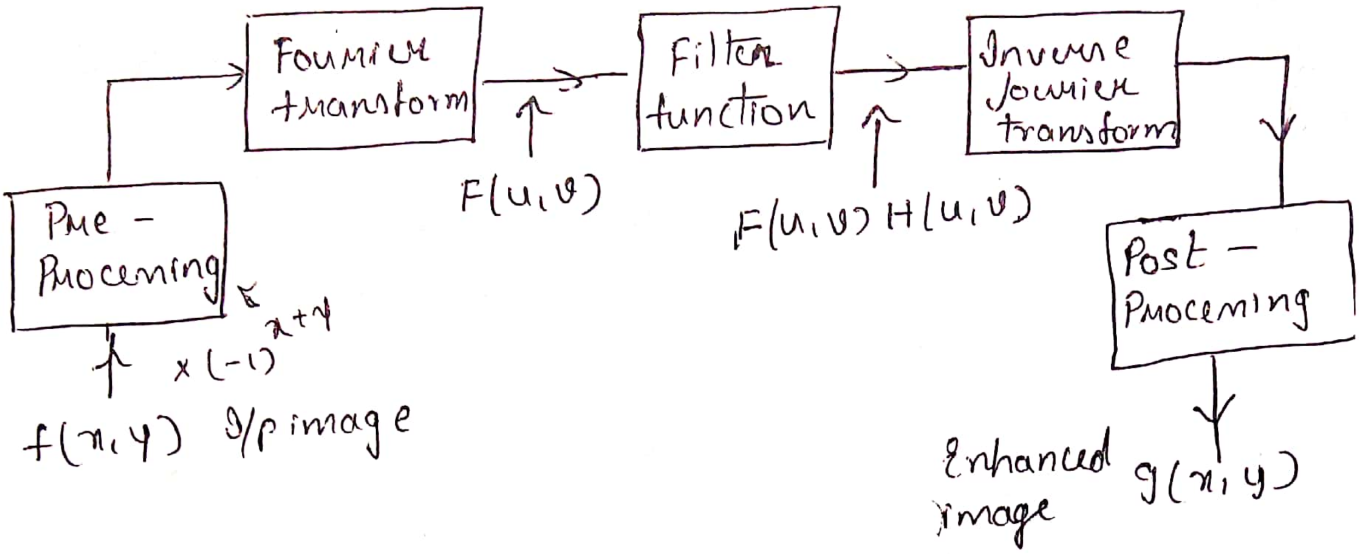
$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

Based on the above expression we observe the following features:

- Each term $F(u,v)$ consists all values of $f(x,y)$ modified by the values of their exponents.
- The slowest varying freq component $(u=v=0)$ corresponds to average gray level of an image.
- The low frequencies nearer to origin correspond to slowly varying components in an image.
- The higher freqs faraway from origin correspond to faster gray level changes in the image. The edges of objects of an image are characterized by these abrupt changes in gray levels such as noise.

BASIC STEPS FOR FILTERING IN FREQ DOMAIN

The block diagram represents the basic steps for filtering in frequency domain.



Pre-processing :- In this stage we multiply the I/p image by $(-1)^{x+y}$ to center the transform.

Fourier transform operation :- compute DFT of the image. $F(u,v)$ is obtained from pre processing.

Filter function :- Multiply the two dimensional DFT by filter function $H(u,v)$. It suppresses certain freq

i.e $G(u,v) = H(u,v) F(u,v)$
enhanced image

Inverse Fourier transform :-

Compute IDFT from $G(u,v)$

So filtered image obtained is

$$F^{-1}[G(u,v)] = g(x,y)$$

Post - Processing :- To compensate pre processing multiply the image by $(-1)^{x+y}$. this process is called post processing.

BASIC FREQUENCY DOMAIN FILTERS :-

Frequency domain filtering is of two types:

(1) Smoothing frequency domain filtering

(2) Sharpening frequency domain filtering

I Smoothing freq domain filtering:-

→ Ideal Low pass filtering (ILPF)

→ Butterworth low pass filter (BLPF)

→ Gaussian Low pass filter (GLPF)

II Sharpening freq domain filtering:-

→ Ideal high pass filter (IHPF)

→ Butterworth high pass filter (BHPF)

→ Gaussian High pass filter (GHPF)

I SMOOTHING FREQ DOMAIN FILTERS:-

(a) Ideal Low pass Filter :- (ILPF)

The low frequencies in Fourier transform are responsible for the general gray-level appearance of an image over smooth areas. Hence ILPF attenuates high frequencies while passing low frequencies. The filtered image will have less sharp details than the original image.

The transfer function of ILPF is given as

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

where D_0 – specified non-negative quantity (cut of freq.)

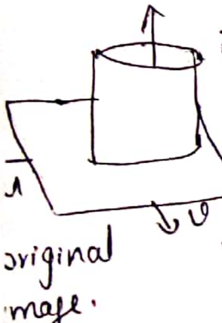
$D(u,v)$ – distance from point (u,v) to the origin of the freq plane.

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

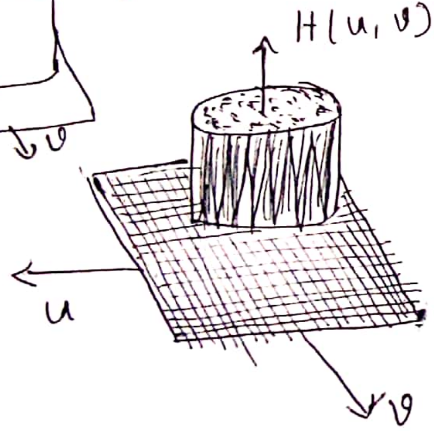
(size of image $M \times N$)

Note:- centre of freq rectangle is at $(u,v) = \left(\frac{M}{2}, \frac{N}{2}\right)$
∵ size of image $M \times N$
transform is also same size.

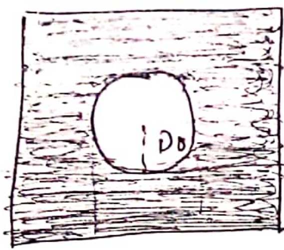
Figs show plot of $H(u,v)$ as an ideal LPF.



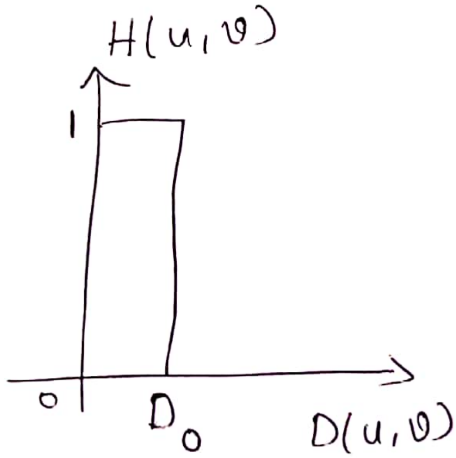
original image.



Perspective plot of ILPF transfer function



Filter displayed as image.



Filter radial cross section

Thus ILPF will pass all freq components inside radius D_0 & all freq components outside D_0 are suppressed.

Disadvantage:- ILPF has ringing problem. Ringing is a visual effect where a series of lines of decreasing intensity lie parallel to the edges. To avoid this BLPF is used.

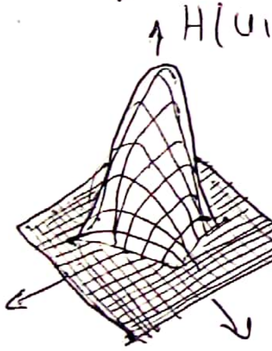
(b) Butterworth Low pass filter :- (BLPF)

The transfer function of a BLPF of order n & with cut-off freq at a distance D_0 from the origin is expressed as

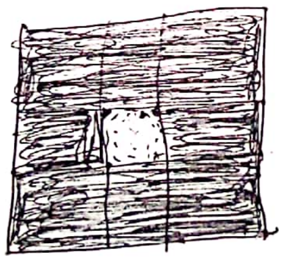
$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

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

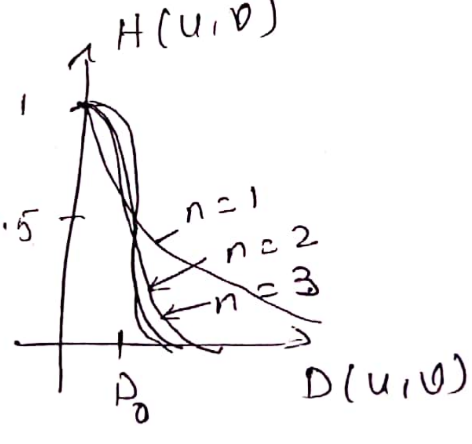
fig shows the plots of BLPF



Perspective plot of BLPF



Filter displayed as image.



Filter radial cross sections of orders 1 through 4.

[Perspective - representation of 3-D objects in 2-D to give right impression of their height, width, depth, position]

→ BLPF has no ringing problem, because it does not exhibit sharp discontinuity like ILPF.

→ As 'n' increases filter becomes sharper with increased ringing in spatial domain.

(c) Gaussian Low pass filter (GLPF) :-

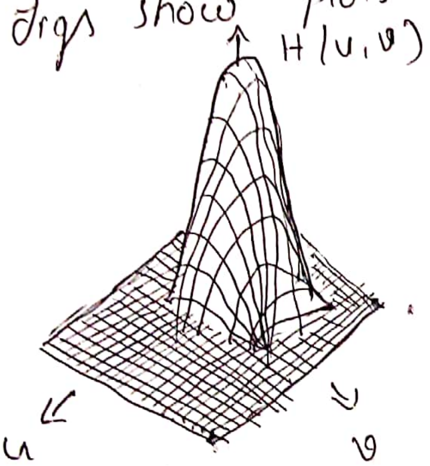
The transfer function of 2D GLPF is

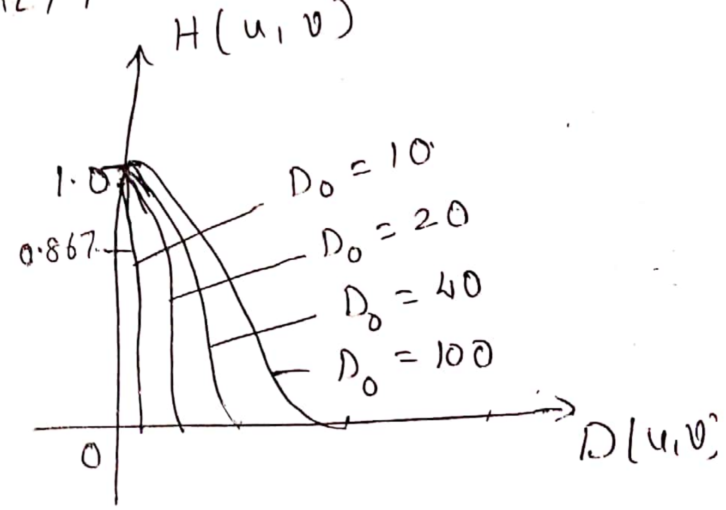
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

The transfer function is controlled by cut-off freq D_0

The GLPF will not have ringing problem.

Figs show plots for GLPF.





II SHARPENING FREQ DOMAIN FILTERS :-

(a) Ideal High Pass Filter :- (IHPF)

A 2D IHPF is one whose transfer function satisfies the relation

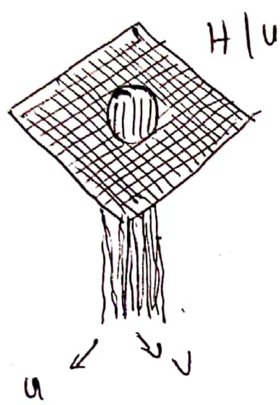
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

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

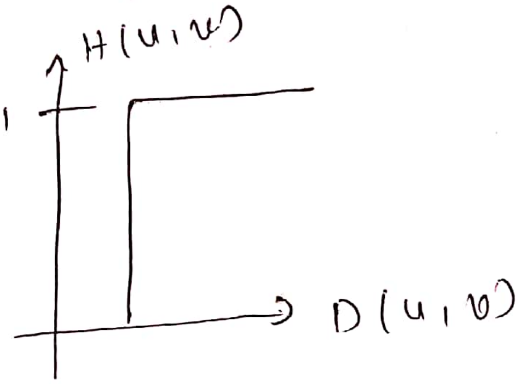
D_0 - cut off freq ; $M \times N$ - size of image.

The IHPF is opposite to ILPF.
IHPF attenuates all the frequency components inside D_0 & allows all the freq components to be passed outside D_0.

The plots are as follows :-



Perspective plot



Radial characteristics

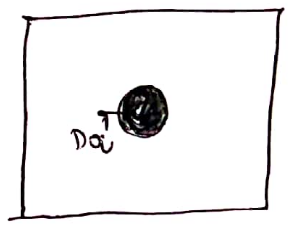
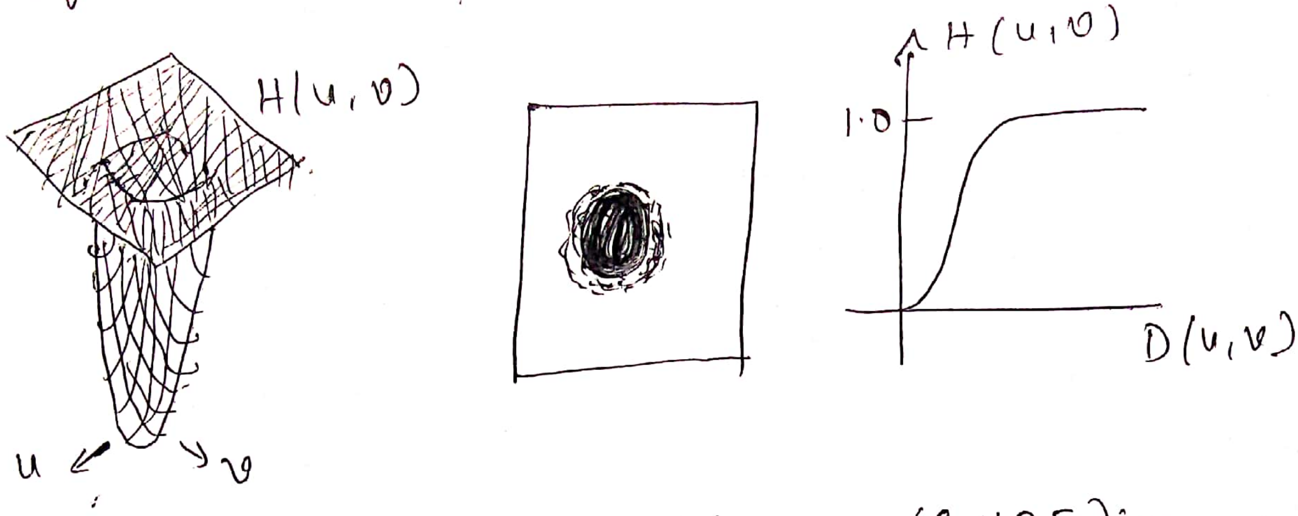


Image Representation

(b) Butterworth High Pass Filter (BHPF) :-
The transfer function of BHPF of order 'n' & with the cut off freq D_0 from the origin is defined by the relation

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

'n' determines sharpness of cut off value & the amount of ringing. The transition into higher values is much smoother in BHPF fig shows the plots



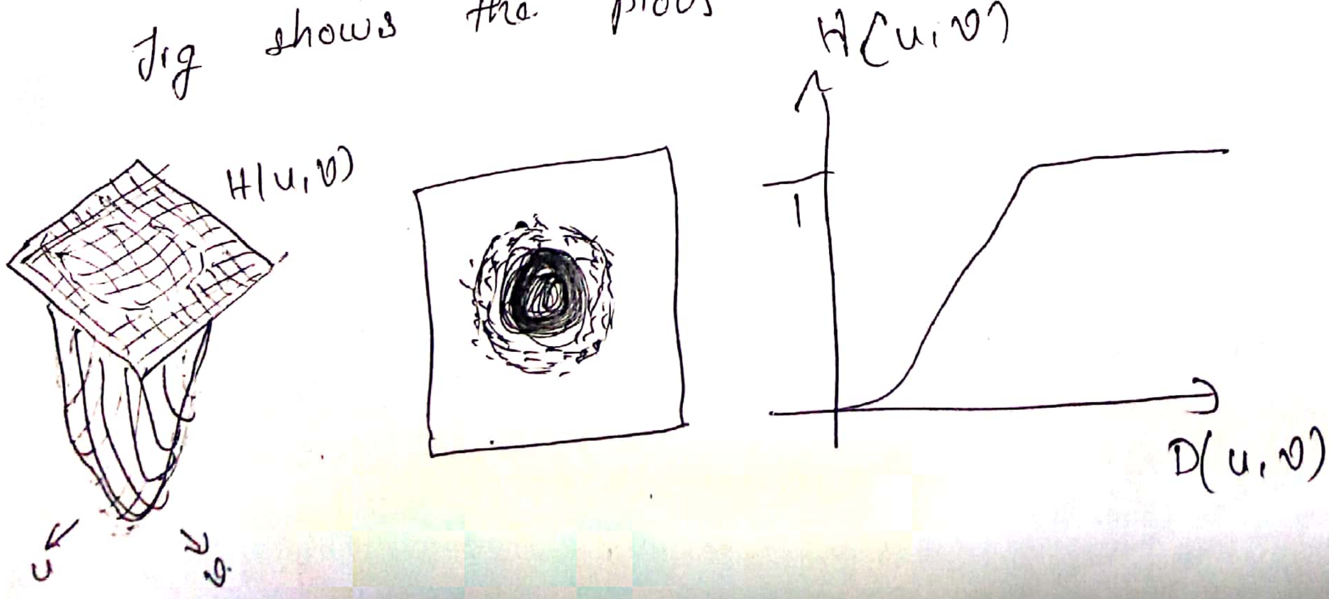
(c) Gaussian High Pass Filtering (GHPF):-

The transfer function of GHPF with cut-off frequency at a distance D_0 from the origin is given by

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

The results obtained are more gradual than IHPF & BHPF.

fig shows the plots



HOMOMORPHIC FILTERING :-

This is a special method of enhancement in which we deal with illumination $i(x,y)$ & reflectance $r(x,y)$ component specially.

Hence an image $f(x,y)$ can be expressed as

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

Applying Fourier transform is not possible because Fourier transform of a product is not product of transforms.

i.e $F[f(x,y)] \neq F[i(x,y)] \cdot F[r(x,y)]$

→ So we use a better mathematical way to define

$$z(x,y) = \ln f(x,y)$$

$$= \ln [i(x,y) \cdot r(x,y)]$$

$$\Rightarrow z(x,y) = \ln i(x,y) + \ln r(x,y)$$

→ Now $F[z(x,y)] = F[\ln i(x,y)] + F[\ln r(x,y)]$

$$Z(u,v) = F_i(u,v) + F_r(u,v)$$

where $F_i(u,v) = F[\ln i(x,y)]$

$F_r(u,v) = F[\ln r(x,y)]$

→ Now multiply this fourier transform image by a filter function $H(u,v)$

$$\therefore \boxed{S(u,v) = Z(u,v)\,H(u,v)}$$

$$S(u,v) = H(u,v)\left[F_i(u,v) + F_n(u,v)\right]$$

$$S(u,v) = H(u,v)\,F_i(u,v) + H(u,v)\cdot F_n(u,v)$$

→ Now take inverse Fourier transform

$$\boxed{s(x,y) = F^{-1}\,S(u,v)}$$

$$= F^{-1}\left[H(u,v)\,F_i(u,v) + H(u,v)\,F_n(u,v)\right]$$

$$= F^{-1}\left[H(u,v)\,F_i(u,v)\right] + F^{-1}\left[H(u,v)\cdot F_n(u,v)\right]$$

$$\Rightarrow \boxed{s(x,y) = i'(x,y) + n'(x,y)}$$

where
$$i'(x,y) = F^{-1}\left[H(u,v)\,F_i(u,v)\right]$$

$$n'(x,y) = F^{-1}\left[H(u,v)\cdot F_n(u,v)\right]$$

→ Now take inverse operation of logarithm applied at beginning of enhanced image.

$$g(x,y) = e^{s(x,y)} = e^{i'(x,y)}\cdot e^{n'(x,y)}$$

$$\Rightarrow g(x,y) = i_0(x,y)\, r_0(x,y)$$

where $i_0(x,y) = e^{i'(x,y)}$

$r_0(x,y) = e^{r'(x,y)}$

there are illumination & reflectance components of i/p image. The illumination component is characterized by slow spatial variations (low freqs) while their reflectance components tends to vary abrupptly (high freqs)

All the above steps are represented in the form of block diagram

$$f(x,y) \rightarrow \boxed{\ln} \rightarrow \boxed{DFT} \rightarrow \boxed{H(u,v)} \rightarrow \boxed{IDFT} \rightarrow \boxed{exp} \rightarrow g(x,y)$$

SELECTIVE FILTERING :-

By using selective filtering only specific band of freqs can be filtered.

- → Band pass filters
- → Band stop (or) Band reject filters
- → Notch filters.

(a) Band Pass filters :- (BPF)

The transfer function of the Band pass filter is given as

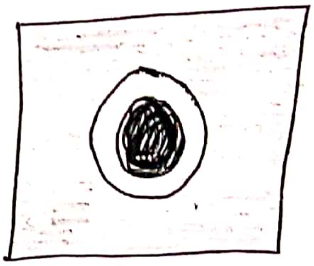
$$H_{BP}(u,v) = \begin{cases} 1 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 0 & \text{otherwise} \end{cases}$$

where D_0 - cut-off freq for 2D BPF

$D(u,v)$ - Distance of point (u,v) from origin

W - width of the band.

The filter displays BPF

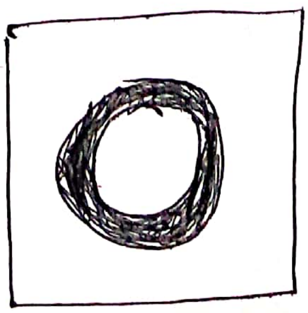


(b) Band reject filters :- (BRF)

The transfer function of BRF is obtained from BPF as

$$H_{BR}(u,v) = 1 - H_{BP}(u,v)$$

$$H_{BR}(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise.} \end{cases}$$



BRF's are effective in removing periodic noise & ringing effect

(c) Notch filters :-

A notch filter is a special form of BRF. Instead of removing entire frequencies it removes selective components.

Notch filters are constructed as products of HPF's.

The general form is

$$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v)\, H_{-k}(u,v)$$

where $H_k(u,v)$ & $H_{-k}(u,v)$ are HPF's whose centers are at (u_k, v_k) & $(-u_k, -v_k)$ resp.

These centres are specified w.r.t center of freq of rectangle $\left(\frac{M}{2}, \frac{N}{2}\right)$

The distance computations for each filter are thus carried out by using expressions

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

& $$D_{-k}(u,v) = \left[\left(u - \frac{M}{2} + u_k\right)^2 + \left(v - \frac{N}{2} + v_k\right)^2\right]^{1/2}$$