## 2.3 SPATIAL DOMAIN FILTERING

Some neighborhood operations work with the values of the image pixels in the neighborhood and the corresponding values of a sub image that has the same dimensions as the neighborhood**. The sub image is called a filter, mask, kernel, template, or window** The values in a filter sub image are referred to as coefficients, rather than pixels.

The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain.

The process consists simply of moving the filter mask from point to point in an image. At each point (x,y), the response of the filter at that point is calculated using a predefined relationship.

**Linear Spatial Filter:**

Linear filter means that the transfer function and the impulse or point spread function of a linear system are inverse Fourier transforms of each other.

For a linear spatial filtering, the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.

Consider a 3 x 3 mask, the response R of linear spatial filtering with the filter mask at a point (x,y) in the image is,

R = w(-1,-1_f(x-1,y+1)+w(-1,0)f(x,y+1)+……+w(1,1)f(x+1,y-1)

Thus the coefficients w(0,0) coincides with image value f(x,y), indicating that the mask is centered at (x,y), when the computation of the sum of products takes place.

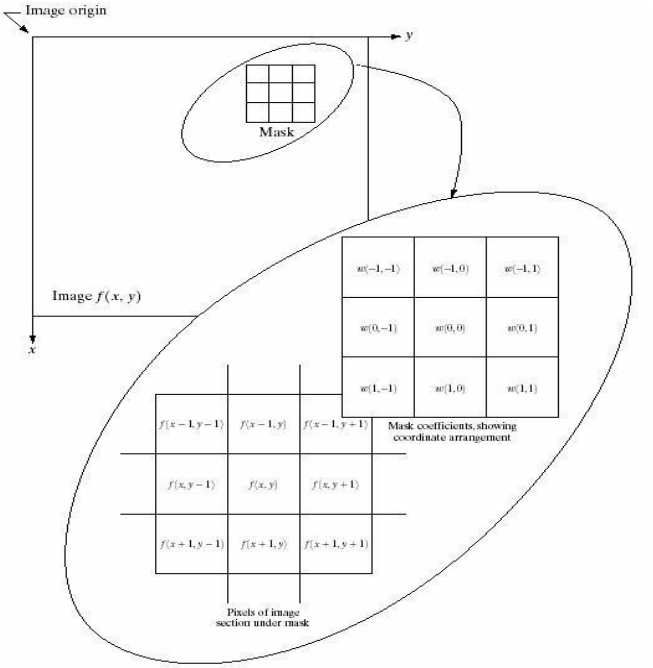
Hence the filter mask size is m x n, where m = 2a+1 and n = 2b+1 and a,b are no-negative integers.In general, the linear filtering of an image f of size M x N with a filter mask of size m x n is given as,

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t), \text{where } ....a = \frac{(m-1)}{2}, b = \frac{(n-1)}{2}$$

Where,

$$x = 0,1,2,....., M-1$$

$$y = 0,1,2,....., N-1$$




From **the general expression, it is seen that the linear spatial filtering is just "convolving a mask with an image". Also the filter mask is called as Convolution mask or Convolution kernel**.

The following shows the 3 x 3 spatial filter mask,

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

$$R = w_1 z_1 + w_2 z_2 + \ldots\ldots + w_9 z_9$$

$$= \sum_{i=1}^{9} w_i z_i$$

**In general**

$$R = w_1 z_1 + w_2 z_2 + \ldots\ldots + w_{mn} z_{mn}$$

$$= \sum_{i=1}^{mn} w_i z_i$$

Where, **w's** are mask coefficients, **z's** are values of the image gray levels corresponding to the mask coefficients and **mn** is the total number of coefficients in the mask.

**Non-Linear Spatial filter:**

This filter is based directly on the values of the pixels in the neighborhood and they don't use the coefficients.These are used to reduce noise.

**Smoothing Spatial Filters or Low pass spatial filtering:**

1. **Smoothing filters are used for blurring and for noise reduction**
2. Blurring is used in preprocessing steps, such as removal of small details from an image prior to object extraction and bridging of small gaps in lines or curves.It includes both linear and non-linear filters

**1.Neighborhood Averaging:**

1. The response of a smoothing linear spatial filter is simply the average of the pixels in the neighborhood of the filter mask,i.e., each pixel is an image is replaced by the average of the gray levels in the neighborhood defined by the filter mask. Hence this filter is called Neighborhood Averaging or low pass filtering.
2. Thus, spatial averaging is used for noise smoothening, low pass filtering and sub sampling of images.
3. This process reduces sharp transition in gray level
4. Thus the averaging filters are used to reduce irrelevant detail in an image.

Consider a 3 x 3 smoothing or averaging filter

$\frac{1}{9}\times$

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

The response will be the sum of gray levels of nine pixels which could cause R to be out of the valid gray level range.

$$R = w_1 z_1 + w_2 z_2 + ..... + w_9 z_9$$

Thus the solution is to scale the sum by dividing R by 9.

$$R = w_1 z_1 + w_2 z_2 + ..... + w_9 z_9$$

$$= \sum_{i=1}^{9} w_i z_i$$

$$= \frac{1}{9} \sum_{i=1}^{9} z_i \text{ , when } .. w_i = 1$$

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

This is nothing but the average of the gray levels if the pixels in the 3 x 3 neighborhood defined by the pixels.
Other spatial low pass filters of various sizes are ,

$\frac{1}{2}$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$\frac{1}{4}$

| | |
|---|---|
| 1 | 1 |
| 1 | 1 |

Thus a m x n mask will have a normalizing constant equal to 1/mn. **A spatial average filter in which all the coefficients are equal is called as a Box Filter.**

**2.Weighted Average Filter: For filtering a M x N image with a weighted averaging filter of size m x m is given by,**

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

Example:

$\frac{1}{16} \times$

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |

Here the pixel at the centre of the mask is multiplied by a higher value than any other, thus giving this pixel more importance.

The diagonals are weighed less than the immediate neighbors of the center pixel. **This reduces the blurring.**

**3. Uniform filtering**

The most popular masks for low pass filtering are masks with all their coefficients positive and equal to each other as for example the mask shown below. Moreover, they sum up to 1 in order to maintain the mean of the image.

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

**4. Gaussian filtering**

The two dimensional Gaussian mask has values that attempts to approximate the continuous function

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{\sigma^2}}$$

. The following shows a suitable integer-valued convolution kernel that approximates a Gaussian with a σ of 1.0.

$$\frac{1}{273}\times\begin{bmatrix}1&4&7&4&1\\4&16&26&16&4\\7&26&41&26&7\\4&16&26&16&4\\1&4&7&4&1\end{bmatrix}$$

**5. Median filtering(Non linear filter)**

Median filtering is the operation that replaces each pixel by the median of the grey level in the neighborhood of that pixel.

Process is replaces the value of a pixel by the median of the gray levels in region $S_{xy}$ of that pixel:

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

**V(m,n) = median{y(m-k,n-l),(k,l)εw}**

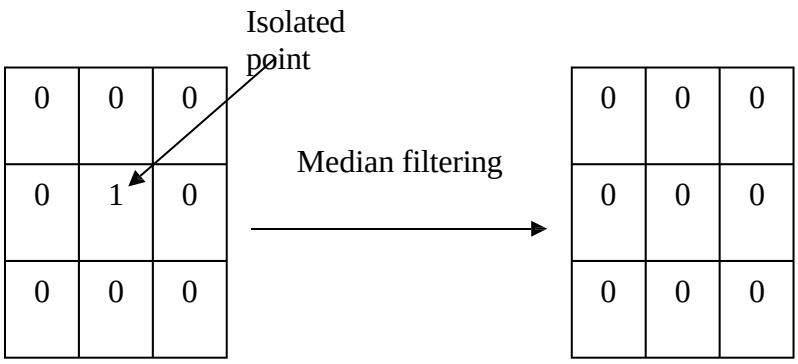
Where w is suitably chosen window. The window size Nw is chosen to be odd.
If Nw is even, then the median is taken as the average of the two values in the middle. Typical window size s are 3x3,5x5,7x7 or 5 point window

**Properties:**

1. Unlike average filtering, **median filtering does not blur t**oo much image details.
**Example:** Consider the example of filtering the sequence below using a
3-pt median filter:16 14 15 12 2 13 15 52 51 50 49
The output of the median filter is:15 14 12 12 13 15 51 51 50
2. **Median filters are non linear filters** because for two sequences $x(n)$ and $y(n)$

$$\text{median}\{x(n)+y(n)\} \neq \text{median}\{x(n)\}+\text{median}\{y(n)\}$$

3. **Median filters are useful for removing isolated lines or points** (pixels) while preserving spatial resolutions.
4. **It works very well on images containing binary (salt and pepper) noise** but performs poorly when the noise is Gaussian.
5. Their performance is also **poor when the number of noise pixels in the window is greater than or half the number of pixels** in the window



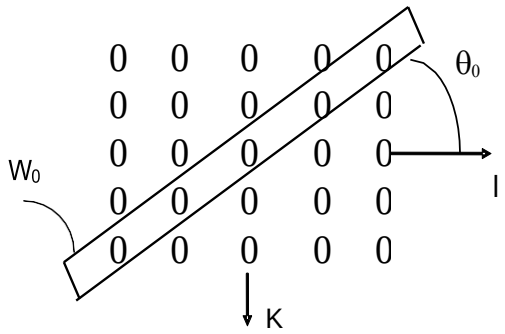

6. ***Directional smoothing***

To protect the edges from blurring while smoothing, a directional averaging filter can be useful. Spatial averages $g(x,y:\theta)$ are calculated in several selected directions (for example could be horizontal, vertical, main diagonals)

$$g(x,y:\theta) = \frac{1}{N_\theta} \sum_{(k,l)\in W_\theta} \sum f(x-k, y-l)$$

and a direction $\theta^*$ is found such that $|f(x,y)-g(x,y:\theta^*)|$ is minimum. (Note that $W_\theta$ is the neighbourhood along the direction $\theta$and $N_\theta$ is the number of pixels within this neighbourhood).Then by replacing $g(x,y:\theta)$ with $g(x,y:\theta^*)$ we get the desired result.

\




**7. <u>Max Filter & Min Filter:</u>**

**The 100th percentile filter is called Maximum filter,** which is used to **find the brightest points** in an image.

$$R = \max\{z_k, k = 1,2,3,....,9\},\ for..3x3matrix$$

**The 0th percentile filter is called Minimum filter,** which is used to **find the darkest points** in an image.

$$R = \min\{z_k, k = 1,2,3,....,9\}\ for.....x3matrix$$

(i.e)Using the 100th percentile results in the so-called max filter, given by

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

This filter is useful for finding the brightest points in an image. Since pepper noise has very low values, it is reduced by this filter as a result of the max selection processing the sub image area $S_{xy}$.

The 0th percentile filter is min filter:

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

This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation

**8. Mean Filters**

This is the simply methods to reduce noise in spatial domain.

1. **Arithmetic mean filter**
2. **Geometric mean filter**
3. **Harmonic mean filter**
4. **Contraharmonic mean filter**

Let $S_{xy}$ represent the set of coordinates in a rectangular sub image window of size mxn, centered at point (x,y).

**1. Arithmetic mean filter**

Compute the average value of the corrupted image g(x,y) in the area defined by $S_{x,y}$.The value of the restored image at any point (x,y)

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

**Note:** Using a convolution mask in which all coefficients have value 1/mn. Noise is reduced as a result of blurring.

**2. Geometric mean filter**

Geometric mean filter is given by the expression

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

**3. Harmonic mean filter**

The harmonic mean filter operation is given by the expression

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

**4. Contraharmonic mean filter**

The contra harmonic mean filter operation is given by the expression

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)_{Q}}$$

Where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise.

**<u>sharpening filters in spatial domain:</u>**

This is used to highlight fine details in an image, to enhance detail that has been blurred and to sharpen the edges of objects

Smoothening of an image is similar to integration process,

Sharpening = 1 / smoothening

Since it is inversely proportional, sharpening is done by differentiation process.

**Derivative Filters:**

1. The derivatives of a digital function are defined in terms of differences.
2. It is used to highlight fine details of an image.

**Properties of Derivative:**

1. For contrast gray level, derivative should be equal to zero
2. It must be non-zero, at the onset of a gray level step or ramp
3. It must be non zero along a ramp [ constantly changes ]

**I$^{st}$ order derivative:**

The first order derivative of one dimensional function f(x) is given as,

$$\frac{\partial f}{\partial x} \cong \left[ f(x+1) - f(x) \right]$$

**II nd order derivative:**

The second order derivative of one dimensional function f(x) is given as,

$$\frac{\partial^2 f}{\partial x^2} \cong \left[ f(x+1) + f(x-1) - 2f(x) \right]$$

Conclusion:

I. First order derivatives produce thicker edges in an image.
II. Second order derivatives have a stronger response to fine details , such as thin lines and isolated points
III. First order derivatives have a stronger response to a gray level step
IV. Second order derivatives produces a double response at step changes in gray level

**Laplacian Operator:**

It is a linear operator and is the simplest isotropic derivative operator, which is defined as,

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

Where,

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

$$\frac{\partial^2 f(x,y)}{\partial^2 x} \approx f(x+1,y) + f(x-1,y) - 2f(x,y) \qquad \text{-------(3)}$$

Sub (2) & (3) in (1), we get,

$$\nabla f^2 \cong \left[ f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \right] \text{----------} (4)$$

The above digital laplacian equation is implemented using filter mask,

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

The digital laplacian equation including the diagonal terms is given as,

$$\nabla f^2 \cong \left[ \begin{array}{l} f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) + f(x+1,y+1) + f(x+1,y-1) \\ + f(x-1,y-1) + f(x-1,y+1) - 8f(x,y) \end{array} \right]$$

the corresponding filter mask is given as,

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

Similarly , two other implementations of the laplacian are,

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | -8 | -1 |
| -1 | -1 | -1 |

The sharpened image using the laplacian operator is given as,

$$g(x, y) \cong \begin{cases} f(x, y) - \nabla^2 f(x, y), \\ f(x, y) + \nabla^2 f(x, y), \end{cases}$$

Substitute equation (4) in (6),

$$g(x, y) \cong f(x, y) - \left[ f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4 f(x, y) \right]$$

$$g(x, y) \cong 5 f(x, y) - \left[ f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) \right]$$

The above digital laplacian equation is implemented using filter mask,

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

This is called as composite laplacian mask.

Substitute equation (5) in (6),

$$g(x, y) \cong f(x, y) - \left[ \begin{array}{l} f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) + f(x+1, y+1) + f(x+1, y-1) \\ + f(x-1, y-1) + f(x-1, y+1) - 8 f(x, y) \end{array} \right]$$

$$g(x,y) \cong 9f(x,y) - \begin{bmatrix} f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)+f(x+1,y+1)+f(x+1,y-1) \\ +f(x-1,y-1)+f(x-1,y+1) \end{bmatrix}$$

The above digital laplacian equation is implemented using filter mask,

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | -8 | -1 |
| -1 | -1 | -1 |

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | A+8 | -1 |
| -1 | -1 | -1 |

This is called the second composite mask.

**Gradient operator:**

For a function f (x, y), the gradient f at co-ordinate (x, y) is defined as the 2-dimesional column vector

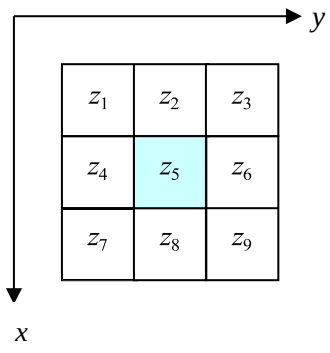
$$\Delta f = \begin{bmatrix} Gx \\ Gy \end{bmatrix}$$

$$= \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

$$\Delta f = \text{mag}(\Delta f) = [Gx^2 + Gy^2]^{1/2} = \{[(\partial f/\partial x)^2 + (\partial f/\partial y)^2]\}^{1/2}$$
$$\cong |Gx| + |Gy|$$

**1. Roberts operator**

consider a $3 \times 3$ mask is the following.



It can be approximated at point $z_5$ in a number of ways. The simplest is to use the difference $(z_8 - z_5)$ in the $x$ direction and $(z_6 - z_5)$ in the $y$ direction. This approximation is known as the **Roberts** operator, and is expressed mathematically as follows.

$$Gx = \left|\frac{\partial f(x, y)}{\partial x}\right| = f(x+1) - f(x) = |z_6 - z_5|$$

$$Gy = \left|\frac{\partial f(x, y)}{\partial y}\right| = f(y+1) - f(y) = |z_8 - z_5|$$

$$\nabla f \cong |Gx| + |Gy|$$

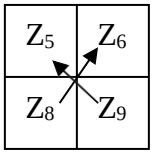
$$\cong |z_6 - z_5| + |z_8 - z_5|$$

| | |
|---|---|
| 1 | -1 |
| 0 | 0 |

| | |
|---|---|
| 1 | 0 |
| -1 | 0 |

Roberts operator

Another approach is to use **Roberts cross differences**
**Take**




This approximation is known as the **Roberts** cross gradient operator, and is expressed mathematically as follows

$$\nabla f \cong |z_9 - z_5| + |z_8 - z_6|$$

The above Equation can be implemented using the following mask

.

| | |
|---|---|
| -1 | 0 |
| 0 | 1 |

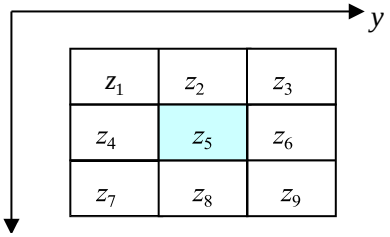
| | |
|---|---|
| 0 | -1 |
| 1 | 0 |

Roberts operator

The original image is convolved with both masks separately and the absolute values of the two outputs of the convolutions are added.

**2. Prewitt operator**

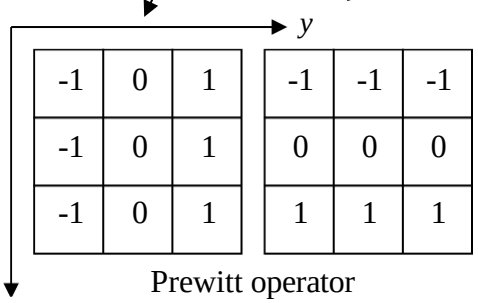
consider a $3 \times 3$ mask is the following.




$$Gx = \left|\frac{\partial f(x,y)}{\partial x}\right| = 3^{rd}row - 1^{st}row = \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right|$$

$$Gy = \left|\frac{\partial f(x,y)}{\partial y}\right| = 3^{rd}col - 1^{st}col = \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right|$$

$$\nabla f \cong |Gx| + |Gy| = |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$




Prewitt operator

This approximation is known as the **Prewitt ator**.

**Basic steps for filtering(image enhancement) in Frequency domain.**

1. Multiply the input image by $(-1)^{x+y}$ to center the transform, image dimensions M x N
2. Compute $F(u, v)$ DFT of the given image ,DC at $M/2, N/2$.
3. Multiply $F(u, v)$ by a filter function $H(u, v)$ to get the FT of the output image,