Imperial College London

Department of Electrical and Electronic Engineering

Digital Image Processing

PART 2

IMAGE ENHANCEMENT

Academic responsible

Dr. Tania STATHAKI

Room 812

Ext. 46229

Email: **t.stathaki@imperial.ac.uk**

http://www.commsp.ee.ic.ac.uk/~tania/

1. Preliminaries

1.1 Spatial domain methods

Suppose we have a digital image which can be represented by a two dimensional random field $f(x,y)$. An image processing operator in the spatial domain may be expressed as a mathematical function $T[\cdot]$ applied to the image $f(x,y)$ to produce a new image $g(x,y)=T[f(x,y)]$ as follows.

$$g(x,y)=T[f(x,y)]$$

The operator T applied on $f(x,y)$ may be defined over:

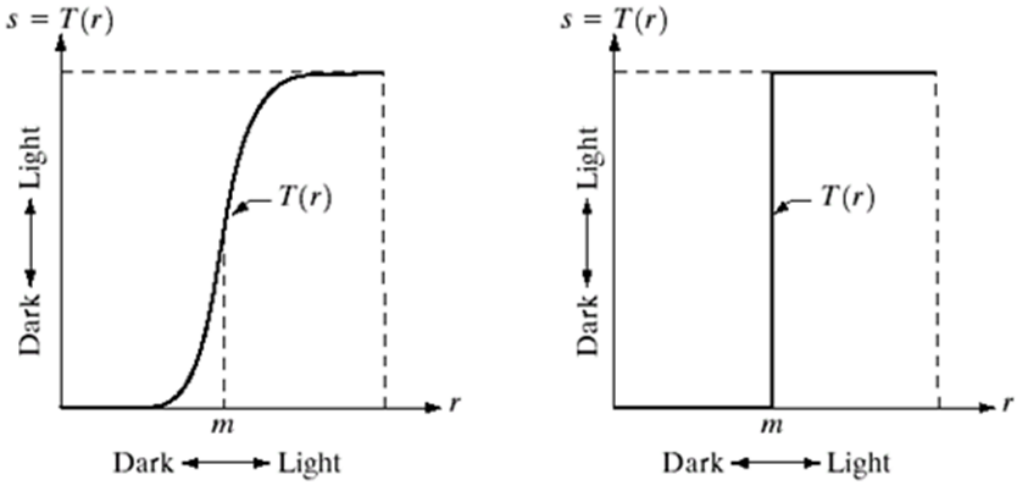
(i) A single pixel (x,y). In this case T is a grey level transformation (or mapping) function.

(ii) Some neighbourhood of (x,y).

(iii) T may operate to a set of input images instead of a single image.

Examples

The intensity transformations shown in the figures below aim at producing an image of higher contrast than the original (figure on the left) or a binary image (figure on the right).



1.2 Frequency domain methods

Let $g(x,y)$ be a desired image formed by the convolution of an image $f(x,y)$ and a linear, position invariant operator $h(x,y)$, that is:

$$g(x,y)=h(x,y)*f(x,y)$$

The following frequency relationship holds:

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

We can select $H(u,v)$ so that the desired image

$$g(x,y)=\Im^{-1}\{H(u,v)F(u,v)\}$$

exhibits some highlighted features of $f(x,y)$. For instance, edges in $f(x,y)$ can be accentuated by using a function $H(u,v)$ that emphasises the high frequency components of $F(u,v)$.

---

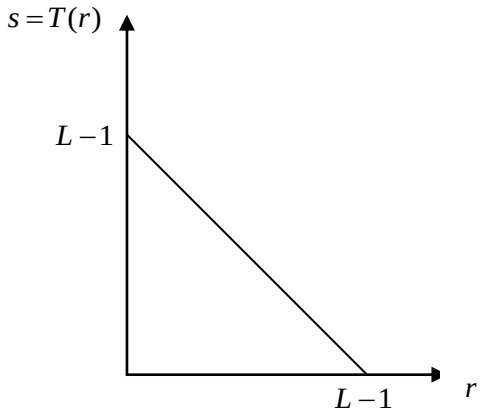
2. Spatial domain: Enhancement by point processing

We are dealing now with image processing methods that are based only on the intensity of single pixels.

2.1 Intensity transformations

2.1.1 Image Negatives

The negative of a digital image is obtained by the transformation function $s = T(r) = L - 1 - r$ shown in the following figure, where L is the number of grey levels. The idea is that the intensity of the output image decreases as the intensity of the input increases. This is useful in numerous applications such as displaying medical images.



2.1.2 Contrast Stretching

Low contrast images occur often due to poor or non uniform lighting conditions, or due to nonlinearity, or small dynamic range of the imaging sensor. In the figure of Example 1 above you have seen a typical contrast stretching transformation.

2.2 Histogram processing. Definition of the histogram of an image.

By processing (modifying) the histogram of an image we can create a new image with specific desired properties.

Suppose we have a digital image of size $N \times N$ with grey levels in the range $[0, L-1]$. The histogram of the image is defined as the following discrete function:

$$p(r_k) = \frac{n_k}{N^2}$$

where

r_k is the kth grey level, $k = 0,1,\ldots,L-1$

n_k is the number of pixels in the image with grey level r_k

N^2 is the total number of pixels in the image

The histogram represents the frequency of occurrence of the various grey levels in the image. A plot of this function for all values of k provides a global description of the appearance of the image.

Question: Think how the histogram of a dark image, a bright image and an image of very low contrast would like. Plot its form in each case.

2.3 Global histogram equalisation

In this section we will assume that the image to be processed has a continuous intensity r that lies within the interval $[0, L-1]$. Let $p_r(r)$ denote the probability density function (pdf) of the variable r. We now apply the following transformation function to the intensity

$$s = T(r) = (L-1)\int_0^r p_r(w)dw, 0 \le r \le L-1 \quad (1)$$

By observing the transformation of equation (1) we immediately see that it possesses the following properties:

(i) $r_2 > r_1 \Rightarrow T(r_2) \ge T(r_1)$, i.e., the function $T(r)$ is increasing with r.

(ii) $T(0) = (L-1)\int_0^0 p_r(w)dw = 0$ and $T(L-1) = (L-1)\int_0^{(L-1)} p_r(w)dw = L-1$. Moreover, if the original image has intensities **only** within a certain range $[r_{\min}, r_{\max}]$ then $T(r_{\min}) = (L-1)\int_0^{r_{\min}} p_r(w)dw = 0$ and $T(r_{\max}) = (L-1)\int_0^{r_{\max}} p_r(w)dw = (L-1)$ since $p_r(r) = 0, r < r_{\min}$ and $r > r_{\max}$. Therefore, the new intensity s takes always all values within the available range $[0, L-1]$.

Suppose that $P_r(r)$, $P_s(s)$ are the cumulative distribution functions (CDF's) of the variables r and s respectively.

Let us assume that the original intensity lies within the values r and $r+dr$ with dr a small quantity. dr can be assumed small enough so as to be able to consider the function $p_r(w)$ constant within the interval $[r, r+dr]$ and equal to $p_r(r)$. Therefore,

$$P_r[r, r+dr] = \int_r^{r+dr} p_r(w)dw \cong p_r(r)\int_r^{r+dr} dw = p_r(r)dr.$$

Now suppose that $s = T(r)$ and $s_1 = T(r+dr)$. The quantity dr can be assumed small enough so as to be able to consider that $s_1 = s + ds$ with ds small enough so as to be able to consider the function $p_s(w)$ constant within the interval $[s, s+ds]$ and equal to $p_s(s)$. Therefore,

$$P_s[s, s+ds] = \int_s^{s+ds} p_s(w)dw \cong p_s(s)\int_s^{s+ds} dw = p_s(s)ds$$

Since $s = T(r)$, $s+ds = T(r+dr)$ and the function of equation (1) is increasing with r, all the values and only the values within the interval $[r, r+dr]$ will be mapped within the interval $[s, s+ds]$. Therefore,

$$P_r[r, r+dr] = P_s[s, s+ds] \Rightarrow p_r(r)dr \overset{r=T^{-1}(s)}{=} p_s(s)ds \Rightarrow p_s(s) = p_r(r)\left.\frac{dr}{ds}\right|_{r=T^{-1}(s)}$$

From equation (1) we see that

$$\frac{ds}{dr} = (L-1)p_r(r)$$

and hence,

$$p_s(s) = \left[p_r(r)\frac{1}{(L-1)p_r(r)} \right]_{r=T^{-1}(s)} = \frac{1}{L-1}, \; 0 \le s \le L-1$$

Conclusion

From the above analysis it is obvious that the transformation of equation (1) converts the original image into a new image with uniform probability density function. This means that in the new image all intensities are present [look at property (iii) above] and with equal probabilities. The whole range of intensities from the absolute black to the absolute white are explored and the new image will definitely have higher contrast compared to the original image.

Unfortunately, in a real life scenario we must deal with digital images. The discrete form of histogram equalisation is given by the relation

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k}\frac{n_j}{N^2} = (L-1)\sum_{j=0}^{k} p_r(r_j),\ 0 \le r_k \le L-1,\ k = 0,1,\ldots,L-1 \quad (2)$$

The quantities in equation (2) have been defined in Section 2.2. To see results of histogram equalisation look at any introductory book on Image Processing.

The improvement over the original image is quite evident after using the technique of histogram equalisation. The new histogram is **not flat** because of the discrete approximation of the probability density function with the histogram function. Note, however, that the grey levels of an image that has been subjected to histogram equalisation are spread out and always reach white. This process increases the dynamic range of grey levels and produces an increase in image contrast.

2.4 Local histogram equalisation

Global histogram equalisation is suitable for overall enhancement. It is often necessary to enhance details over small areas. The number of pixels in these areas my have negligible influence on the computation of a global transformation, so the use of this type of transformation does not necessarily guarantee the desired local enhancement. The solution is to devise transformation functions based on the grey level distribution – or other properties – in the neighbourhood of every pixel in the image. The histogram processing technique previously described is easily adaptable to local enhancement. The procedure is to define a square or rectangular neighbourhood and move the centre of this area from pixel to pixel. At each location the histogram of the points in the neighbourhood is computed and a histogram equalisation transformation function is obtained. This function is finally used to map the grey level of the pixel centred in the neighbourhood. The centre of the neighbourhood region is then moved to an adjacent pixel location and the procedure is repeated. Since only one new row or column of the neighbourhood changes during a pixel-to-pixel translation of the region, updating the histogram obtained in the previous location with the new data introduced at each motion step is possible quite easily. This approach has obvious advantages over repeatedly computing the histogram over all pixels in the neighbourhood region each time the region is moved one pixel location. Another approach often used to reduce computation is to utilise non overlapping regions, but this methods usually produces an undesirable checkerboard effect.

2.5 Histogram specification

Suppose we want to specify a particular histogram shape (not necessarily uniform) which is capable of highlighting certain grey levels in the image.

Let us suppose that:

$p_r(r)$ is the original probability density function

$p_z(z)$ is the desired probability density function

Suppose that histogram equalisation is first applied on the original image r

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

Suppose that the desired image z is available and histogram equalisation is applied as well

$$v = G(z) = (L-1)\int_0^z p_z(w)dw$$

$p_s(s)$ and $p_v(v)$ are both uniform densities and they can be considered as identical. Note that the final result of histogram equalisation is independent of the density inside the integral. So in equation $v = G(z) = (L-1)\int_0^z p_z(w)dw$ we can use the symbol s instead of v.

The inverse process $z = G^{-1}(s)$ will have the desired probability density function. Therefore, the process of histogram specification can be summarised in the following steps.

(i) We take the original image and equalise its intensity using the relation $s = T(r) = (L-1)\int_0^r p_r(w)dw$.

(ii) From the given probability density function $p_z(z)$ we specify the probability distribution function $G(z)$.

(iii) We apply the inverse transformation function $z = G^{-1}(s) = G^{-1}[T(r)]$

---

3. Spatial domain: Enhancement in the case of many realisations of an image of interest available

3.1 Image averaging

Suppose that we have an image $f(x,y)$ of size $M \times N$ pixels corrupted by noise $n(x,y)$, so we obtain a noisy image as follows.

$$g(x,y) = f(x,y) + n(x,y)$$

For the noise process $n(x,y)$ the following assumptions are made.

(i) The noise process $n(x,y)$ is ergodic.

(ii) It is zero mean, i.e., $E\{n(x,y)\} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y) = 0$

(ii) It is white, i.e., the autocorrelation function of the noise process defined as $R[k,l] = E\{n(x,y)n(x+k,y+l)\} = \frac{1}{(M-k)(N-l)}\sum_{x=0}^{M-1-k}\sum_{y=0}^{N-1-l} n(x,y)n(x+k,y+l)$ is zero, apart for the pair $[k,l] = [0,0]$.

Therefore, $R[k,l] = \frac{1}{(M-k)(N-l)}\sum_{x=0}^{M-1-k}\sum_{y=0}^{N-1-l} n(x,y)n(x+k,y+l) = \sigma^2_{n(x,y)}\delta(k,l)$ where $\sigma^2_{n(x,y)}$ is the variance of noise.

Suppose now that we have L different noisy realisations of the same image $f(x,y)$ as $g_i(x,y) = f(x,y) + n_i(x,y)$, $i = 0,1,\ldots,L$. Each noise process $n_i(x,y)$ satisfies the properties (i)-(iii) given above. Moreover, $\sigma^2_{n_i(x,y)} = \sigma^2$. We form the image $\bar{g}(x,y)$ by averaging these L noisy images as follows:

$$\bar{g}(x,y) = \frac{1}{L}\sum_{i=1}^{L} g_i(x,y) = \frac{1}{L}\sum_{i=1}^{L}(f(x,y) + n_i(x,y)) = f(x,y) + \frac{1}{L}\sum_{i=1}^{L} n_i(x,y)$$

Therefore, the new image is again a noisy realisation of the original image $f(x,y)$ with noise $n(x,y) = \frac{1}{L}\sum_{i=1}^{L} n_i(x,y)$.

The mean value of the noise $n(x,y)$ is found below.

$$E\{n(x,y)\} = E\{\frac{1}{L}\sum_{i=1}^{L} n_i(x,y)\} = \frac{1}{L}\sum_{i=1}^{L} E\{n_i(x,y)\} = 0$$

The variance of the noise $n(x,y)$ is now found below.

$$\sigma^2_{n(x,y)} = E\{n^2(x,y)\} = E\left\{\left(\frac{1}{L}\sum_{i=1}^{L} n_i(x,y)\right)^2\right\} = \frac{1}{L^2}E\left\{\left(\sum_{i=1}^{L} n_i(x,y)\right)^2\right\}$$

$$= \frac{1}{L^2}E\{(\sum_{i=1}^{L} n_i^{\,2}(x,y))\} + \frac{1}{L^2}E\{(\sum_{\substack{i=1\\ i\neq j}}^{L}\sum_{j=1}^{L}(n_i(x,y)n_j(x,y))\} = \frac{1}{L^2}\sum_{i=1}^{L}E\{n_i^{\,2}(x,y)\} + \frac{1}{L^2}\sum_{\substack{i=1\\ i\neq j}}^{L}\sum_{j=1}^{L}E\{n_i(x,y)n_j(x,y)\}$$

$$= \frac{1}{L^2}\sum_{i=1}^{L}\sigma^2 + 0 = \frac{1}{L}\sigma^2$$

Therefore, we have shown that image averaging produces an image $\bar{g}(x,y)$, corrupted by noise with variance less than the variance of the noise of the original noisy images. Note that if $L \to \infty$ we have $\sigma^2_{n(x,y)} \to 0$, the resulting noise is negligible.

---

4. Spatial domain: Enhancement in the case of a single image

4.1 Spatial masks

Many image enhancement techniques are based on spatial operations performed on local neighbourhoods of input pixels.
The image is usually convolved with a finite impulse response filter called spatial mask. The use of spatial masks on a digital image is called spatial filtering.
Suppose that we have an image $f(x,y)$ of size N^2 and we define a neighbourhood around each pixel. For example let this neighbourhood to be a rectangular window of size 3×3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

If we replace each pixel by a weighted average of its neighbourhood pixels then the response of the linear mask for the pixel z_5 is $\sum_{i=1}^{9} w_i z_i$. We may repeat the same process for the whole image.

4.2 Lowpass and highpass spatial filtering

A 3×3 spatial mask operating on an image can produce (a) a smoothed version of the image (which contains the low frequencies) or (b) it can enhance the edges and suppress essentially the constant background information. The behaviour is basically dictated by the signs of the elements of the mask.
Let us suppose that the mask has the following form

a	b	c
d	1	e
f	g	h

To be able to estimate the effects of the above mask with relation to the sign of the coefficients a,b,c,d,e,f,g,h, we will consider the equivalent one dimensional mask

d	1	e

Let us suppose that the above mask is applied to a signal $x(n)$. The output of this operation will be a signal $y(n)$ as $y(n) = dx(n-1) + x(n) + ex(n+1) \Rightarrow Y(z) = dz^{-1}X(z) + X(z) + ezX(z) \Rightarrow$

$Y(z) = (dz^{-1} + 1 + ez)X(z) \Rightarrow \frac{Y(z)}{X(z)} = H(z) = dz^{-1} + 1 + ez$. This is the transfer function of a system that produces the above input-output relationship. In the frequency domain we have $H(e^{j\omega}) = d\exp(-j\omega) + 1 + e\exp(j\omega)$.

The values of this transfer function at frequencies $\omega = 0$ and $\omega = \pi$ are:

$H(e^{j\omega})\Big|_{\omega=0} = d + 1 + e$

$H(e^{j\omega})\Big|_{\omega=\pi} = -d + 1 - e$

If a lowpass filtering (smoothing) effect is required then the following condition must hold

$H(e^{j\omega})\Big|_{\omega=0} \geq H(e^{j\omega})\Big|_{\omega=\pi} = d + e \geq 0$

If a highpass filtering effect is required then

$H(e^{j\omega})\Big|_{\omega=0} \leq H(e^{j\omega})\Big|_{\omega=\pi} = d + e \leq 0$

The most popular masks for lowpass filtering are masks with all their coefficients positive and for highpass filtering, masks where the central pixel is positive and the surrounding pixels are negative or the other way round.

4.3 Popular techniques for lowpass spatial filtering

4.3.1 Uniform filtering

The most popular masks for lowpass 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.

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

1	1	1
1	1	1
1	1	1

4.3.2 Gaussian filtering

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

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

In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. The following shows a suitable integer-valued convolution kernel that approximates a Gaussian with a σ of 1.0.

$\frac{1}{273}\times$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4

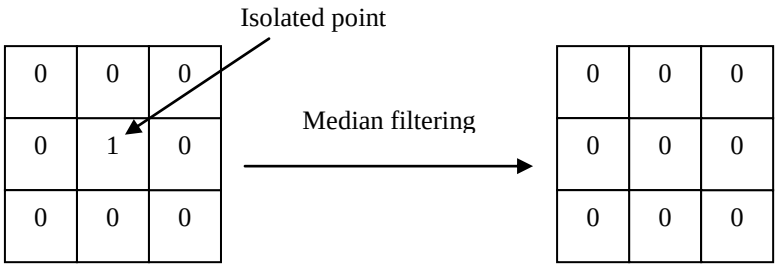
4.3.3 Median filtering

The median m of a set of values is the value that possesses the property that half the values in the set are less than m and half are greater than m. Median filtering is the operation that replaces each pixel by the median of the grey level in the neighbourhood of that pixel.

Median filters are non linear filters because for two sequences $x(n)$ and $y(n)$

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

Median filters are useful for removing isolated lines or points (pixels) while preserving spatial resolutions. They perform very well on images containing binary (**salt and pepper**) noise but perform poorly when the noise is Gaussian. 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 (**why?**)



4.3.4 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 θ^* is found such that $\left|f(x,y)-g(x,y:\theta^*)\right|$ is minimum. (Note that W_θ is the neighbourhood along the direction θ and N_θ 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.

4.3.5 High Boost Filtering

A high pass filtered image may be computed as the difference between the original image and a lowpass filtered version of that image as follows:

(Highpass part of image)=(Original)-(Lowpass part of image)

Multiplying the original image by an amplification factor denoted by A, yields the so called high boost filter:

(Highboost image)= (A) (Original)-(Lowpass)= $(A-1)$ (Original)+(Original)-(Lowpass)

= $(A-1)$ (Original)+(Highpass)

The general process of subtracting a blurred image from an original as given in the first line is called ***unsharp masking***. A possible mask that implements the above procedure could be the one illustrated below.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{9}\times \begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

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

The high-boost filtered image looks more like the original with a degree of edge enhancement, depending on the value of A.

4.4 Popular techniques for highpass spatial filtering. Edge detection using derivative filters

4.4.1 About two dimensional high pass spatial filters

An edge is the boundary between two regions with relatively distinct grey level properties. The idea underlying most edge detection techniques is the computation of a local derivative operator. The magnitude of the first derivative calculated within a neighbourhood around the pixel of interest, can be used to detect the presence of an edge in an image.

The gradient of an image $f(x,y)$ at location (x,y) is a vector that consists of the partial derivatives of $f(x,y)$ as follows.

$$\nabla \underline{f}(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

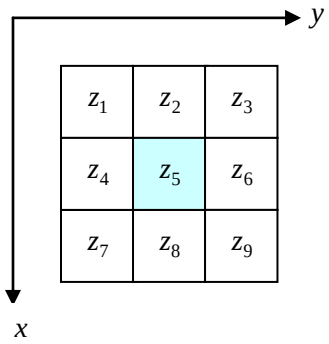
The magnitude of this vector, generally referred to simply as the gradient ∇f is

$$\nabla f(x,y) = \text{mag}(\nabla \underline{f}(x,y)) = \left[\left(\frac{\partial f(x.y)}{\partial x} \right)^2 + \left(\frac{\partial f(x,y)}{\partial y} \right)^2 \right]^{1/2}$$

Common practice is to approximate the gradient with absolute values which is simpler to implement as follows.

$$\nabla f(x,y) \cong \left| \frac{\partial f(x,y)}{\partial x} \right| + \left| \frac{\partial f(x,y)}{\partial y} \right| \qquad (1)$$

Consider a pixel of interest $f(x,y) = z_5$ and a rectangular neighbourhood of size $3\times 3 = 9$ pixels (including the pixel of interest) as shown below.



4.4.2 Roberts operator

Equation (1) can be approximated at point z_5 in a number of ways. The simplest is to use the difference $(z_5 - z_8)$ in the x direction and $(z_5 - z_6)$ in the y direction. This approximation is known as the **Roberts** operator, and is expressed mathematically as follows.

$$\nabla f \cong |z_5 - z_8| + |z_5 - z_6| \tag{2}$$

Another approach for approximating (1) is to use cross differences

$$\nabla f \cong |z_5 - z_9| + |z_6 - z_8| \tag{3}$$

Equations (2), (3) can be implemented by using the following masks. The original image is convolved with both masks separately and the absolute values of the two outputs of the convolutions are added.

1	0
-1	0

1	-1
0	0

Roberts operator

1	0
0	-1

0	1
-1	0

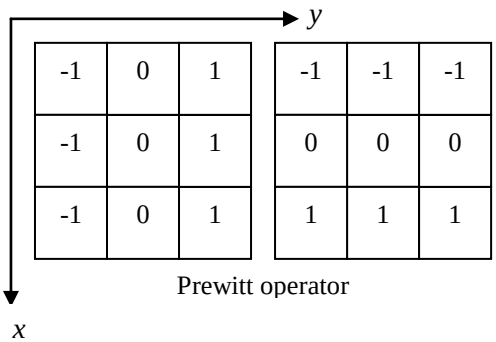
Roberts operator

4.4.3 Prewitt operator

Another approximation of equation (1) but using now a 3×3 mask is the following.

$$\nabla f \cong |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)| \tag{4}$$

This approximation is known as the **Prewitt** operator. Equation (4) can be implemented by using the following masks. Again, the original image is convolved with both masks separately and the absolute values of the two outputs of the convolutions are added.



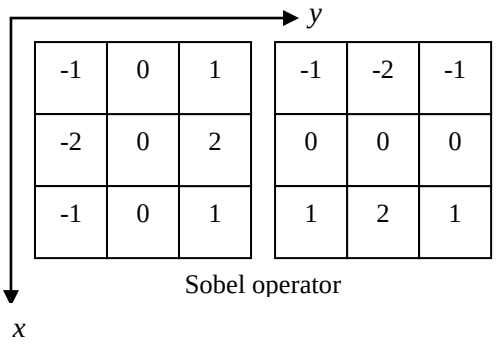
Prewitt operator

4.4.4 Sobel operator. Definition and comparison with the Prewitt operator

The most popular approximation of equation (1) but using a 3×3 mask is the following.

$$\nabla f \cong \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right| \quad (5)$$

This approximation is known as the **Sobel** operator.



Sobel operator

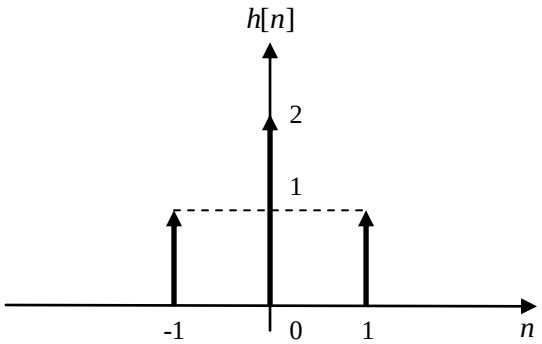
If we consider the left mask of the Sobel operator, this causes differentiation along the y direction. A question that arises is the following: What is the effect caused by the same mask along the x direction? If we isolate the following part of the mask

1
2
1

and treat it as a one dimensional mask, we are interested in finding the effects of that mask. We will therefore, treat this mask as a one dimensional impulse response $h[n]$ of the form

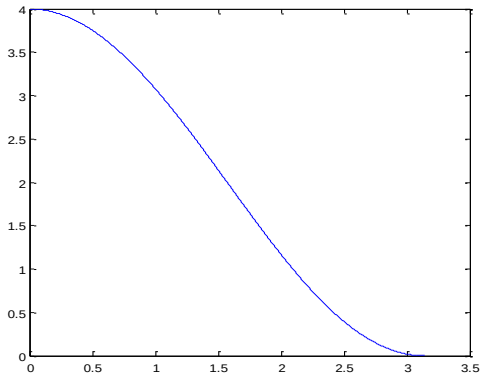
$$h[n] = \begin{cases} 1 & n = -1 \\ 2 & n = 0 \\ 1 & n = 1 \\ 0 & \text{otherwise} \end{cases}$$

or

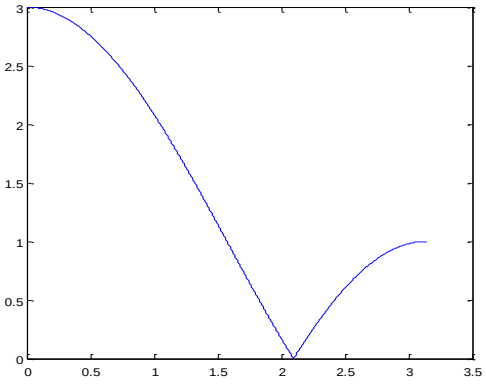


The above response applied to a signal $x[n]$ yields a signal $y[n] = x[n-1] + 2x[n] + x[n+1]$ or in z-transform domain $Y(z) = (z^{-1} + 2 + z)X(z) \Rightarrow Y(j\omega) = 2(\cos\omega + 1)X(j\omega)$. Therefore, $h[n]$ is the

impulse response of a system with transfer function $H(j\omega)=2(\cos\omega+1)=|H(j\omega)|$ shown in the figure below for $[0,\pi]$. This is a lowpass filter type of response. Therefore, we can claim that the Sobel operator has a differentiation effect along one of the two directions and a smoothing effect along the other direction.



The same analysis for the Prewitt operator would give $Y(z)=(z^{-1}+1+z)X(z)$ $\Rightarrow Y(j\omega)=(2\cos\omega+1)X(j\omega)\Rightarrow |H(j\omega)|=|2\cos\omega+1|$ shown in the figure below for $[0,\pi]$. This response looks "strange" since it decreases up to the point $|2\cos\omega+1|=0\Rightarrow\cos\omega=-0.5$ and then starts increasing.



Based on the above analysis it is stated in the literature that the Sobel operator have the advantage of providing both a differencing a smoothing effect while Prewitt does not. However, if you implement both operators you cannot see any visual difference.

4.4.5 Laplacian operator

The **Laplacian** of a 2-D function $f(x,y)$ is a second order derivative defined as

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

In practice it can be also implemented using a 3x3 mask as follows (**why?**)

$$\nabla^2 f=4z_5-(z_2+z_4+z_6+z_8)$$

The main disadvantage of the Laplacian operator is that it produces double edges (**why?**).

To see real images where all the above algorithms have been applied look at any Image Processing book.

The material of these lecture notes on **Image Enhancement** was mainly taken from the following books, and modified for the needs of this course.

[1] ***Digital Image Processing*** by R. C. Gonzales and R. E. Woods, Addison-Wesley Publishing Company, 1992.

[2] ***Fundamentals of Digital Image Processing*** by A. K. Jain, Prentice Hall, 1989.