Spatial and Gray level resolution:

Spatial resolution:

Is the smallest discernible detail in an image. it is determined by the sampling process. The Spatial resolution of a digital image reflects the amount of details that one can see in the image.

Gray level resolution:

It refers to the smallest discernible change in gray level. it is determined by the quantization process. the number of gray level is an integer power of 2. The most common number is 8-bits, however, 16 bits is used in some applications where enhancement of specific gray level ranges is necessary.

Decreasing the gray level resolution of a digital image may result in what is known as false countouring. This effect is caused by the use of an isufficient number of gray levels in smooth areas of digital image

The pixel values of the following 5x5 image are represented by 8-bit integers:

$$f = \begin{matrix} 123 & 162 & 200 & 147 & 93 \\ 137 & 157 & 165 & 232 & 189 \\ 151 & 155 & 152 & 141 & 130 \\ 205 & 101 & 100 & 193 & 115 \\ 250 & 50 & 75 & 88 & 100 \end{matrix}$$

Determine (f) with gray level resolution of 2^K, when $K=5$ and $K=3$?

Dividing the image by 2 will reduce its gray level resolution by one bit.

Hence to reduce the gray level resolution from 8-bit to 5-bit, we have to reduce 3-bits.

8-bits - 5-bits = 3bits will be reduced.

Thus we divide the 8-bit image by (2^3) to get the following 5-bit image.

$$f = \begin{matrix} 15 & 20 & 25 & 18 & 11 \\ 17 & 19 & 20 & 29 & 23 \\ 18 & 19 & 19 & 17 & 16 \\ 25 & 12 & 12 & 24 & 14 \\ 31 & 6 & 9 & 10 & 12 \end{matrix}$$

Similarly to obtain 3-bit, we divide the 8-bit image by $(32)\ 2^5$.

$$f = \begin{matrix} 3 & 5 & 6 & 4 & 2 \\ 4 & 4 & 5 & 7 & 5 \\ 4 & 4 & 4 & 4 & 4 \\ 6 & 3 & 3 & 6 & 3 \\ 7 & 1 & 2 & 2 & 3 \end{matrix}$$

Zooming.

نعوضها بين عناصر الصورة عشان اكثار التفاصيل الازاحة

Zooming may be viewed as oversampling. it is scaling of an image area A of $r \times c$ pixels by a factor S

مع حفاظ على الدقة المكانية

while maintaning spatial resolution.

Zooming: increasing the number of pixels in an image so that the image appears larger.

Zooming requires two steps:

1- Creation of new pixel locations

تعيين

2- assignment of gray level to these new locations.

There are many methods of gray level assignments, for example nearest neighbor interpolation and bilinear interpolation.

① Nearest neighbor interpolation.

is performed by repeating pixel values, thus creating checkerboard effect. pixel replication is used to increase the size of an image an integer number of times.

The example shows 8-bit image Zooming by 2X (2 times) using neighbor interpolation.

$$\begin{bmatrix} 69 & 50 & 80 \\ 45 & 60 & 66 \\ 30 & 55 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix}$$

Original image — image with rows expanded — image with rows and columns expanded

② Bilinear interpolation.

is performed by finding linear interpolation between adjacent pixels, thus creating blurring effect. this can be done by finding the average gray value between two pixels and use that as the pixel value between those two. we can do this for the rows first, and then we take take that result and expand the columns in the same waye. The example below shows 8-bit image zooming by 2x (2 times) using bilinear interpolation.

$$\begin{bmatrix} 69 & 50 & 80 \\ 45 & 60 & 66 \\ 30 & 55 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 59 & 50 & 65 & 80 \\ 45 & 52 & 60 & 63 & 66 \\ 30 & 42 & 55 & 67 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 59 & 50 & 65 & 80 \\ 57 & 55 & 55 & 64 & 73 \\ 45 & 52 & 60 & 63 & 66 \\ 37 & 47 & 57 & 65 & 73 \\ 30 & 42 & 55 & 67 & 80 \end{bmatrix}$$

Original image — image with rows expanded — image with rows and columns expanded

Shrinking

Shrinking may be viewed as undersampling. image shrink is performed by row-column deletion. to shrink an image by one half, we delet every other row and column.

$$\begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 50 & 80 \\ 45 & 60 & 66 \\ 30 & 55 & 80 \end{bmatrix}$$

Original image — image with rows deleted — image with rows and columns deleted

Point processing of images

A general image processing operator is a function that takes one or more input images and produces an output image.

image transform can be seen as:

* point operators.
* Neighborhood operators.

in a digital image, point = pixel. point processing transforms a pixel's value as function of its value alone, it does not depend on the values of the pixel's neighbors.

point processing of images.

1- Brightness and contrast adjustment.

These may be specific several ways:

A- Gain and Basic Modification:

pixel values are transformed as:

$$V_{out} = g \times V_{in} + b.$$

The parameters $g > 0$ and b are often called the gain and bias parameters; sometimes these parameters are said to control contrast and brightness respectively.

2- Mean and standard deviation adjustment.

in this case, the gain g and bias b are computed from the specified mean and standard deviation, μ_{new} and σ_{new}, and from the actual mean and standard deviation of data, μ_{data} and σ_{data} which must be computed by:

$$g = \frac{\sigma_{new}}{\sigma_{data}}$$

$$b = \mu_{new} - g \times \mu_{data}.$$

For example: Given the image.

3	2	10
15	6	2
11	3	4

$\sigma_{new} = 3$

$\mu_{new} = 7$

standard deviation(data) $\sigma_{data} =$

$$\sqrt{(\Sigma(x_i)^2/n) - (\Sigma x_i/n)^2}$$

$$= \sqrt{(9+4 \cdots +16)/9 - (56/9)^2}$$

$$\sigma_{data} = \sqrt{58.2 - 42.9} = 3.9 \simeq 4$$

$\mu_{data} = 6.2$

$$g = \frac{\sigma_{new}}{\sigma_{data}} = \frac{3}{4} = 1$$

$$b = \mu_{new} - g \times \mu_{data} = 7 - 1 \times 5 = 2$$

2 - increase brightness.

The effect of this transformation is to separate out the low pixel value and compress the high so that details in dark area are made more visible at the expense of details in the bright.

To increase the brightness use the equation:

$$J(r,c) = \begin{cases} I(r,c) + g & \text{if } I(r,c) + g < 256 \\ 255 & \text{if } I(r,c) + g > 255 \end{cases}$$

where:

J is the image output.

I is the image input, $g \geq 0$.

3- Decrease brightness.

To decrease brightness use the equation:

$$J(r,c) = \begin{cases} 0 & \text{if } I(r,c) - g < 0. \\ I(r,c) - g & \text{if } I(r,c) \geq 0. \end{cases}$$

where:

J is the image output.

I is the image input and $g \geq 0$.

image algebra.

There are two categories of algebraic operations applied to images:

* Arithmetic.
* Logic.

These operations are performed on a pixel by pixel basis between two or more images, except for the NOT logic operation which requires only one image.

Arithmetic operations can be classified into four classe, Addition, subtraction, division and multiplication, while AND, OR and NOT are the logic operations.

To apply the arithmetic operations to two images, we simply operate on corresponding pixel values. for example to add image I_1 and I_2 to create I_3:

$$\underset{I_1}{\begin{matrix} 3 & 4 & 7 \\ 3 & 4 & 5 \\ 2 & 4 & 6 \end{matrix}} + \underset{I_2}{\begin{matrix} 6 & 6 & 6 \\ 4 & 2 & 6 \\ 3 & 5 & 5 \end{matrix}} = \underset{I_3}{\begin{matrix} 9 & 10 & 13 \\ 7 & 6 & 11 \\ 5 & 9 & 11 \end{matrix}}$$

Addition is used to combine the information in two images. Applications include development of image restoration algorithm for ~~modeoteling~~ modeling additive noise.

subtraction of two images is often used to detect motion. For example, in a ~~scene~~ scene when nothing has changed, the image resulting from the subtraction is filled with zeros (black image). If something has changed in the scene, subtraction produces a non zero result at the location of movement.

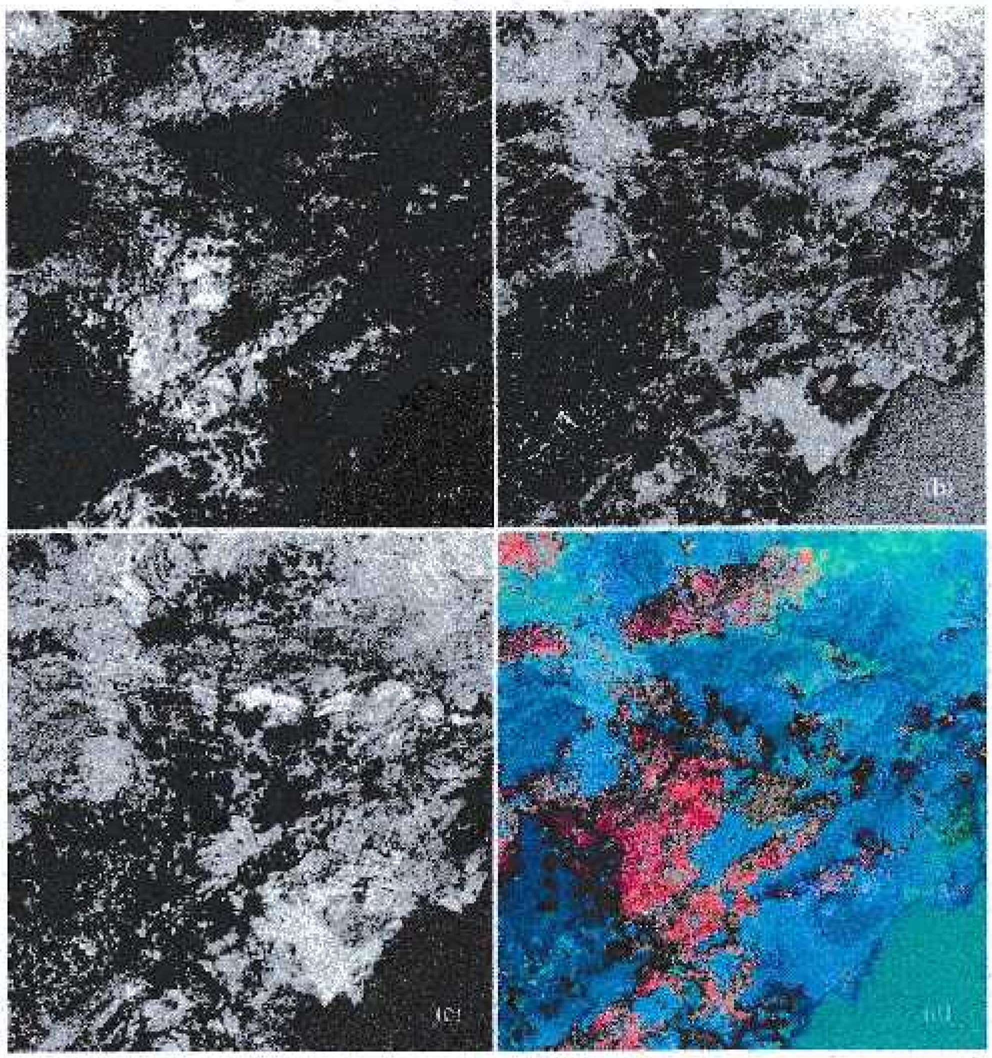


Figure 3.1 Difference images of a Landsat TM image: (a) *TM*3 – *TM*1 highlights red features often associated to iron oxides; (b) *TM*4 – *TM*3 detects the diagnostic 'red edge' features of vegetation; (c) *TM*5 – *TM*7 enhances the clay mineral absorption features in SWIR spectral range; and (d) the colour composite of *TM*3 – *TM*1 in red, *TM*4 – *TM*3 in green and *TM*5 – *TM*7 in blue highlights iron oxide, vegetation and clay minerals in red, green and blue colours

Multiplication and division are used to adjust the brightness of an image. Multiplying the pixel values by a number greater than one will brighten the image, and dividing the pixel values by a factor greater than one will darken the image.

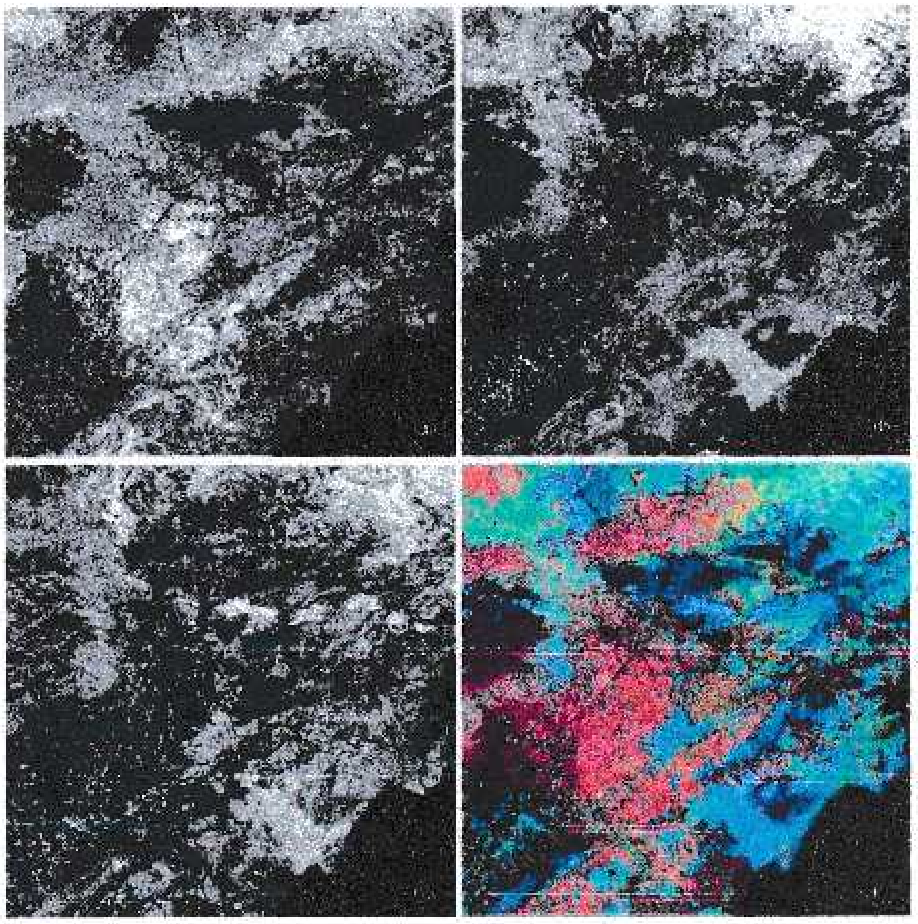


Figure 3.4 Ratio images and ratio colour composite: (a) the ratio image of *TM3/TM1*; (b) the ratio image of *TM4/TM3*; (c) the ratio image of *TM5/TM7*; and (d) the ratio colour composite of *TM5/TM7* in blue, *TM4/TM3* in green and *TM3/TM1* in red

In logical (Boolean) operators are used to compare conditional states of a pixel (TRUE or FALSE). The three logical operators are AND, OR and NOT.

$\cap$	AND	&	image 1 & image 2
$\cup$	OR	|	image 1|image 2
not	NOT	~	~ image 1

A True state is usually encoded as a 1 or nonzero integer, while a false state is usually encoded as a 0.

For example:

image 1

$$\begin{matrix} 3 & 2 & 0 \\ 1 & 2 & 3 \\ 7 & 1 & 0 \end{matrix}$$

image 2

$$\begin{matrix} 2 & 3 & 5 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \end{matrix}$$

image 1 greater than 2 AND image 2 greater than 1.

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

image 1 greater than 2 OR image 2 greater than 1.

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{matrix}$$

Neighbourhood processing (spatial filters).

Neighbourhood processing is an extension of point processing, where a function is applied to a neighbourhood of each pixel. The idea is to move an "mask" over the given image.

The combination of mask and function is called a "filter".

if the function of the new gray value is a linear function of all the gray values in the mask, then the filter is called a linear filter.

Steps performing spatial filter:

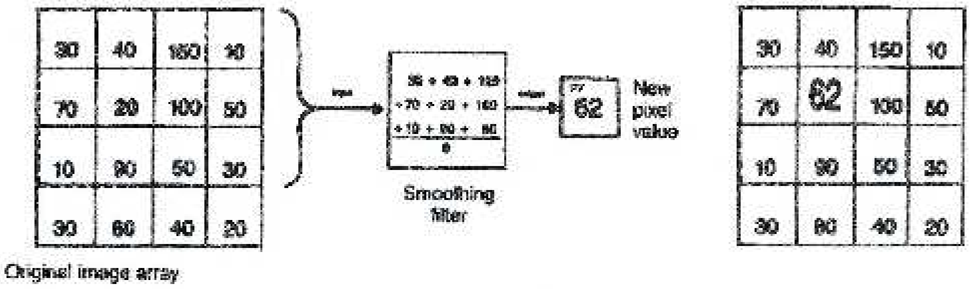
1. position the mask over the image.
2. The filtered image is "convolution" of the original image with the mask.

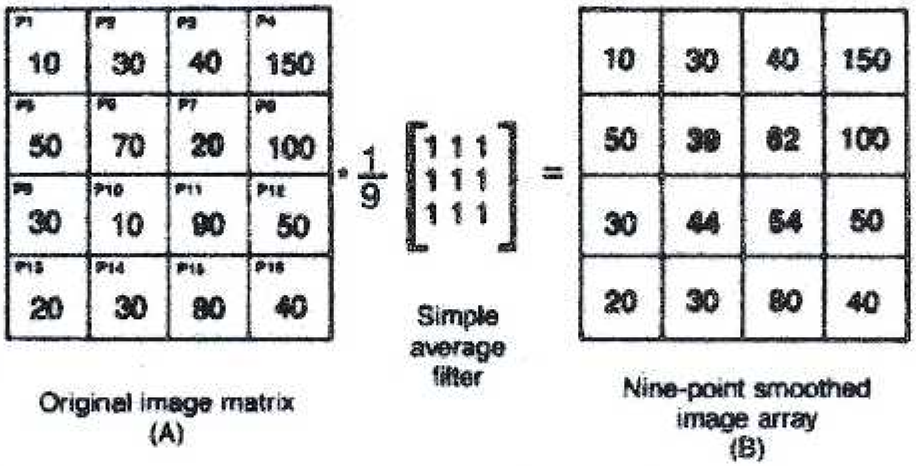
   So, if $f(x,y)$ denotes the original image and $w(x,y)$ is the mask, then their convolution is:

$$f'(x,y) = (w * f)_{(x,y)}$$

$$= \sum_{s=-a}^{a} \sum_{t=-b}^{b} \underbrace{w(s,t)}_{\text{mask}} \underbrace{f(x+s, y+t)}_{\text{pixel}}.$$

3. add up all the products.


30 40 150 10
70 20 100 50
10 90 50 30
30 80 40 20
Original image array
Smoothing filter
62
New pixel value
30 40 150 10
70 62 100 50
10 90 50 30
30 80 40 20


P1 10
P2 30
P3 40
P4 150
P5 50
P6 70
P7 20
P8 100
P9 30
P10 10
P11 90
P12 50
P13 20
P14 30
P15 80
P16 40
* 1/9
1 1 1
1 1 1
1 1 1
=
10 30 40 150
50 39 62 100
30 44 54 50
20 30 80 40
Simple average filter
Original image matrix
(A)
Nine-point smoothed image array
(B)

Averaging Filter. (mean filters).

it is one important linear filter that use (3×3 , 5×5 , 7×7 ----) mask and take the average of all values within the mask. This value become the gray level of the corresponding pixel in the new image.

apply mask average for the image?

0	1	10	100	102
5	4	11	102	108
4	7	100	103	105
5	8	9	10	100
10	50	50	57	60
10	51	52	55	61

$$M(4) = \frac{1}{9}(0+1+10+5+4+11+4+7+100) = \frac{142}{9} = 15.7 = 16$$

$$M(11) = \frac{1}{9}(1+10+100+4+11+102+7+100+103) = \frac{438}{9} = 48.6 = 49$$

$$M(102) = \frac{1}{9}(10+100+102+11+102+108+100+103+105) = \frac{741}{9} = 82$$

$$M(7) = \frac{1}{9}(5+4+11+4+7+100+5+8+9) = \frac{153}{9} = 17$$

$$M(100) = \frac{1}{9}(4+11+102+7+100+103+8+9+10) = \frac{354}{9} = 39.3 = 39$$

$$M(103) = \frac{1}{9}(11+102+108+100+103+105+9+10+100) = \frac{648}{9} = 72$$

M_1

$M(8) = 27$

$M(9) = 44$

$M(10) = 66$

$M(50) = 27$

$M(50) = 38$

$M(57) = 50$

The output image is:

0	1	10	100	102
5	16	49	82	108
4	17	39	72	105
5	27	44	66	100
10	27	38	50	60
10	51	52	55	61

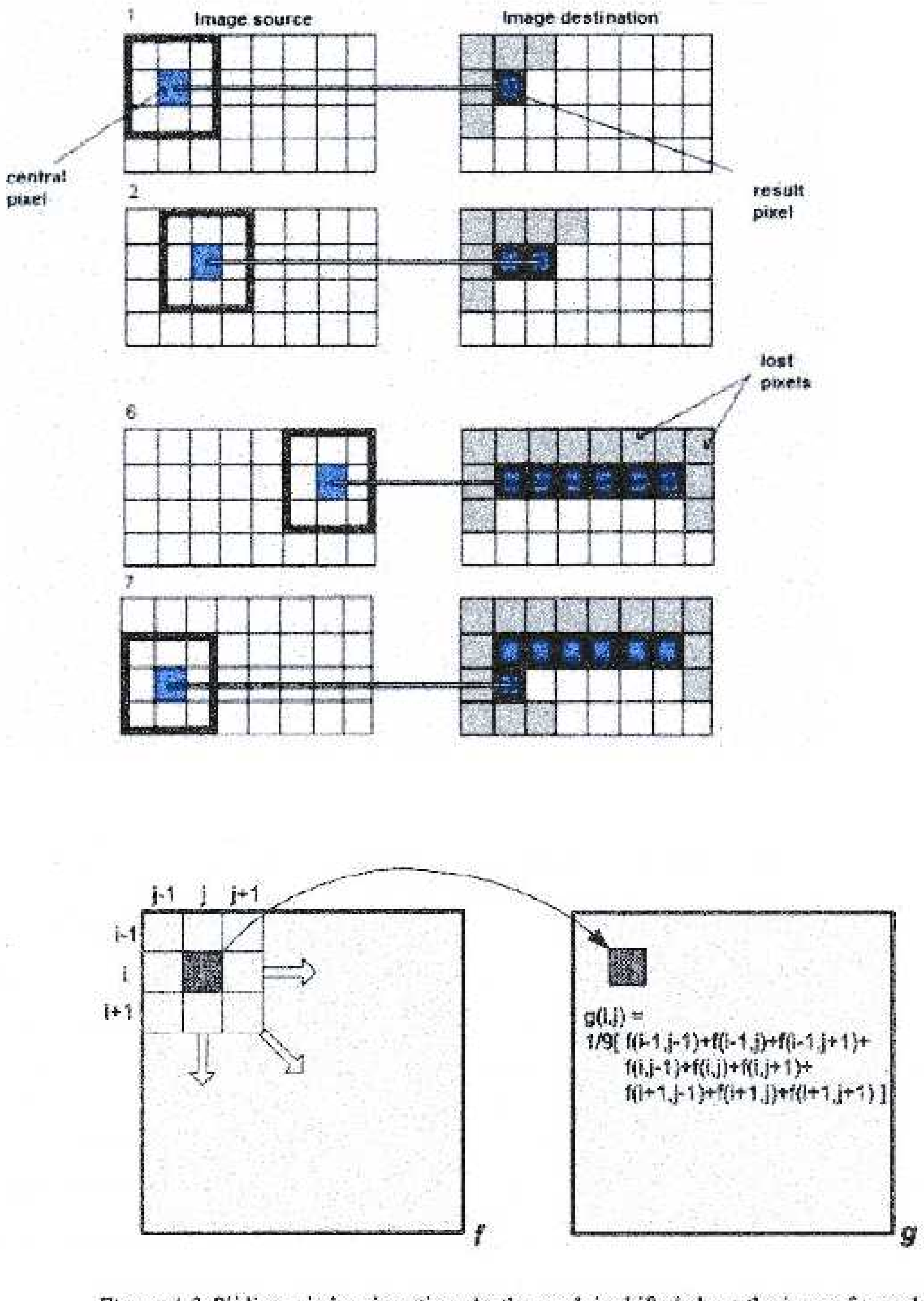


Figure 4-3. Sliding window in action. As the mask is shifted about the image for each position *(i,j)*, the intensity *g(i,j)* is set to the average of *f(i,j)*'s nine surrounding neighbors.