

Point coordinates on digital image

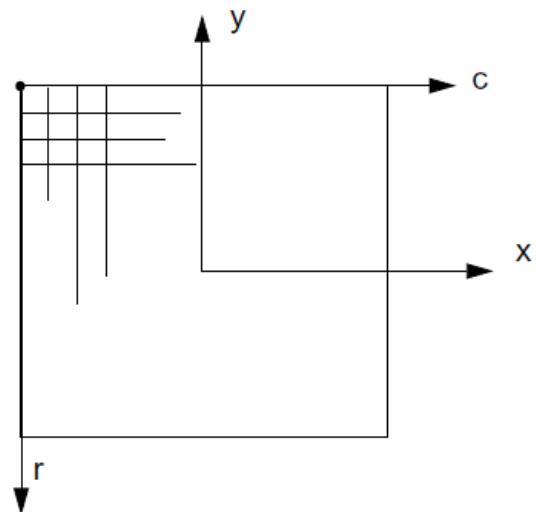
A digital image is made up of a series of pixels (picture elements). The dimensions of a digital image are expressed in terms of its pixels, for instance 1500 by 1200, where the first number indicates the total number of image columns and the second number is the total number of image rows. Image resolution is measured by number of image pixels per inch or size of image pixel usually expressed in millimeters or microns.

Pixel coordinate system

The file coordinates of a digital image are defined in a pixel coordinate system, shown by c- and r- axes in the figure. A pixel coordinate system is a coordinate system with its origin in the upper-left corner of the image, the c-axis pointing to the right, the r-axis pointing downward, and the units in pixels. These file coordinates (c, r) can also be thought of as the pixel column number and pixel row number, respectively.

Image coordinate system

An image coordinate system is defined as a 2D coordinate system existing on the image plane with its origin usually at the image center, shown by x- and y- axes in the figure. On aerial photographs, the axes are determined by fiducial lines joining opposite fiducial marks. If the principal point does not coincide with the fiducial center, the system is shifted to bring its origin at that point. Image coordinates are used to describe positions on the photo plane. Image coordinate units are usually millimeters or microns.



Example-1

The column and row coordinates of a point p on an image having a 600 dpi resolution are 995 and 480 pixels, respectively. The total image size is 1600 columns by 1400 rows. Compute the point coordinates in mm in the photographic coordinate system, related to the image center.

Solution:

Coordinates of image center in the pixel coordinate system (c_o, r_o)

$$= (1600/2, 1400/2) = 800 \text{ c}, 700 \text{ r}$$

Coordinates of point p related to the image center in pixels:

$$x_p = c_p - c_o = 995 - 800 = 195 \text{ pixel}$$

$$y_p = -(r_p - r_o) = -(480 - 700) = 220 \text{ pixel}$$

$$\text{Pixel size in mm} = \frac{1 \text{ inch}}{600} = \frac{25.4 \text{ mm}}{600} = 0.042 \text{ mm}$$

Coordinates of point p related to the image center in mm:

$$x_p = 195 \times 0.042 = 8.19 \text{ mm}$$

$$y_p = 220 \times 0.042 = 9.24 \text{ mm}$$

Example-2

An image has a resolution of 900 dpi (dot per inch). Calculate the equivalent ground sample distance, given that the camera focal length is 152 mm and the flying height above ground is 1920m.

Solution:

f=152 mm and H=1920 m.

$$\text{Scale} = \frac{f}{H} = \frac{152/1000}{1920} = 1:12632$$

$$\text{Pixel size} = \frac{1 \text{ inch}}{900} = \frac{25.4 \text{ mm}}{900} = 0.028 \text{ mm}$$

$$\text{Ground sample distance} = \text{Pixel size} \times \frac{1}{\text{Scale}} = 0.028 \times 12632 = 353.7 \text{ mm} = 35.37 \text{ cm}$$

Example-3

Given an image having a 600 dpi resolution. The total image size is 1800 columns by 1500 rows. Compute coordinates of an image p in the pixel coordinate system, given that its photographic x and y coordinates are -12.42 mm and -25.36 mm, respectively.

Solution:

Coordinates of image center in the pixel coordinate system (c_o, r_o)

$$= (1800/2, 1500/2) = 900 \text{ c}, 750 \text{ r}$$

$$\text{Pixel size} = \frac{1 \text{ inch}}{600} = \frac{25.4 \text{ mm}}{600} = 0.042 \text{ mm}$$

Coordinates of point p related to image center in pixels:

$$c_p = -\frac{12.42 \text{ mm}}{0.042 \text{ mm/pixel}} = -295.7 \text{ pixel}$$

$$r_p = -\frac{25.36 \text{ mm}}{0.042 \text{ mm/pixel}} = -603.8 \text{ pixel}$$

Coordinates of point p in pixels related to pixel coordinate system:

$$c_p = 900 - 295.7 = 604.3 \text{ pixel}$$

$$r_p = 750 + 603.8 = 1353.8 \text{ pixel}$$

Coordinate system transformation

Example-4

Given the measured coordinates of the four fiducials of an aerial photograph in the Pixel coordinate system and the corresponding calibrated coordinates in the photo coordinate system. Compute the parameters of (a) similarity transformation and (b) affine transformation necessary to convert from the Pixel coordinate system to the photo coordinate system.

Fiducial No.	measured coordinates(pixel)		calibrated coordinates (mm)	
	x' (c)	y' (r)	x	y
1	48.15	2699.98	-113.002	0.008
2	5353.12	2699.13	112.991	0.036
3	2700.21	47.62	0.008	112.998
4	2700.56	5352.74	-0.016	-113

Solution:

(a) Similarity transformation

$$x = a x' - b y' + c$$

$$y = b x' + a y' + d$$

in which a,b,c,d are similarity transformation parameters.

The known coordinates of the four points in the two systems give 8 equations in 4 unknowns leading to

$$\begin{matrix}
 \begin{bmatrix} 48.15 & 2699.98 & 1 & 0 \\ -2699.98 & 48.15 & 0 & 1 \\ 5353.12 & 2699.13 & 1 & 0 \\ -2699.13 & 5353.12 & 0 & 1 \\ 2700.21 & 47.62 & 1 & 0 \\ -47.62 & 2700.21 & 0 & 1 \\ 2700.56 & 5352.74 & 1 & 0 \\ -5352.74 & 2700.56 & 0 & 1 \end{bmatrix} & \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} & = & \begin{bmatrix} -113.002 \\ 0.008 \\ 112.991 \\ 0.036 \\ 0.008 \\ 112.998 \\ -0.016 \\ -113 \end{bmatrix} & + & \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} \\
 A & X & = & L & + & V
 \end{matrix}$$

The solution vector $X = (A^T A)^{-1} (A^T L)$

$$X = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0.043 \\ 0.000 \\ -115.030 \\ 115.040 \end{bmatrix}$$

$$V = A X - L = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} 0.006 \\ 0.009 \\ 0.006 \\ -0.006 \\ -0.014 \\ -0.001 \\ 0.002 \\ -0.002 \end{bmatrix}$$

(b) Affine transformation

$$x = a x' + b y' + e$$

$$y = c x' + d y' + f$$

in which a, b, c, d, e, f are affine transformation parameters.

The known coordinates of the four points in the two systems give 8 equations in 6 unknowns leading to

$$\begin{matrix}
 \begin{bmatrix} 48.15 & -2699.98 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 48.15 & -2699.98 & 1 \\ 5353.12 & -2699.13 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5353.12 & -2699.13 & 1 \\ 2700.21 & -47.62 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2700.21 & -47.62 & 1 \\ 2700.56 & -5352.74 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2700.56 & -5352.74 & 1 \end{bmatrix} &
 \begin{bmatrix} a \\ b \\ e \\ c \\ d \\ f \end{bmatrix} &
 = &
 \begin{bmatrix} -113.002 \\ 0.008 \\ 112.991 \\ 0.036 \\ 0.008 \\ 112.998 \\ -0.016 \\ -113 \end{bmatrix} &
 + &
 \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} \\
 A & & & X & = & L & + & V
 \end{matrix}$$

The solution vector $X = (A^T A)^{-1} (A^T L)$

$$X = (A^T A)^{-1} (A^T L)$$

$$X = \begin{bmatrix} a \\ b \\ e \\ c \\ d \\ f \end{bmatrix} = \begin{bmatrix} 0.043 \\ 0.000 \\ -115.030 \\ -0.000 \\ 0.043 \\ 115.030 \end{bmatrix}$$

$$V = A X - L = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} 0.006 \\ 0.002 \\ 0.006 \\ 0.002 \\ -0.006 \\ -0.002 \\ -0.006 \\ -0.002 \end{bmatrix}$$

Digital image matching

Reconstruction of a 3-D object space model is one of the most important processes in photogrammetry. The key drive of this process is to identify corresponding features from stereo pairs of images. The process is commonly named as stereo matching, or digital stereo matching at using digital imagery. The basic criterion for the matching is to measure the similarity between

the overlapping image features. Different matching techniques have different measures of similarity. The most prominent matching problems are relief, occlusions, discontinuities and non-linear radiometric differences.

One of the most popular matching techniques is the intensity-based matching, where a patch of one image (template) is compared with a patch of the same size (matching) within a search window in the other image. The similarity criterion is found by either searching for the maximum cross-correlation coefficient, or by minimizing the gray level differences using least-squares adjustment.

Correlation matching depends on the assumption that there will be a uniquely defined maximum in the correlation function. A variety of operational systems are based on this approach or on a modified version of it. The main advantage of the approach is the rather simple implementation. The arbitrary definition of window size, the correlation function that may not yield a unique maximum are among the problems at applying the approach.

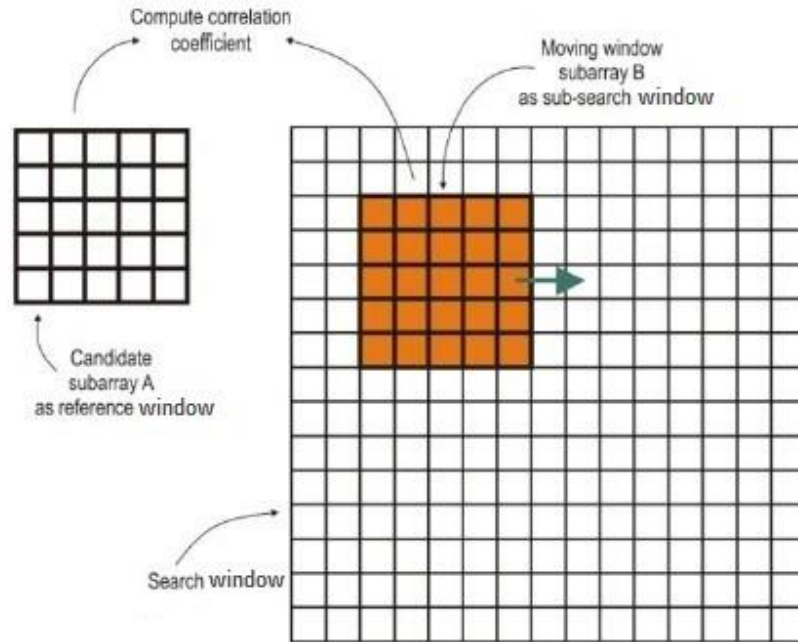
On the other hand, the least-squares (LS) matching might prove to be better, since it accounts for radiometric distortion using linear transformation, and for geometric distortion employing geometric transformation. Usually, an affine transformation is adopted.

Another matching approach is the feature-based matching, where features in both images are detected and matched. Image features can be points, lines and shapes. The process of matching is usually on the basis of similarity of the feature attributes or descriptions such as orientation, gradient, shape, etc. Then, a consistency check is performed to discard the erroneous matches.

Several approaches that use either feature-based matching or hybrid matching, which combines intensity-based and feature-based matching methods, have been developed. Consequently, developments in procedures such as automatic image orientation and automatic aerial triangulation have become possible. However, there are still problems that require image matching to be applied with some human interactions. Optimal window sizes, the need of good approximations, and reliability of matching results are some of the problems of intensity-based matching.

Correlation matching

Here, the similarity of a template, or a reference window, with every position of the matching window within the search space is measured using the cross correlation coefficient. The position that yields the maximum correlation coefficient is to be determined.



The correlation coefficient is computed by the following equation, using digital numbers from subarrays A and B

$$\rho = \frac{\sum_{i=1}^m \sum_{j=1}^n [(A_{ij} - \bar{A})(B_{ij} - \bar{B})]}{\sqrt{[\sum_{i=1}^m \sum_{j=1}^n (A_{ij} - \bar{A})^2][\sum_{i=1}^m \sum_{j=1}^n (B_{ij} - \bar{B})^2]}}$$

Where ρ is the correlation coefficient; m and n are the numbers of rows and columns, respectively, in the subarrays A and B; A_{ij} is the digital number from subarray A at row i and column j ; \bar{A} is the average of all digital numbers in subarray A; B_{ij} is the digital number from subarray B at row i and column j ; and \bar{B} is the average of all digital numbers in subarray B.

The correlation coefficient can range from -1 to +1, with +1 indicating perfect correlation (an exact match). A coefficient of -1 indicates negative correlation, coefficient values near zero indicate no correlation.

Due to factors such as image noise, perfect correlation is extremely rare. Generally a threshold value such as 0.7 is chosen and if the correlation coefficient exceeds that value the subarrays are assumed to match.

Example-5 (1-D correlation matching)

Given that [1,2,9,9,8,2,1] and [1,2,2,1,1,3,8,8,6,1,1,1] are template and search windows having one dimensional pattern. Compute normalized correlation coefficient between the template and all possible subwindows in the search window.

Solution:

Let the template A = [1,2,9,9,8,2,1]

All possible subwindows to be matched with the template A can be given as follows:

B₁=[1,2,2,1,1,3,8] , B₂=[2,2,1,1,3,8,8] , B₃=[2,1,1,3,8,8,6] , B₄=[1,1,3,8,8,6,1] , B₅=[1,3,8,8,6,1,1] , B₆=[3,8,8,6,1,1,1]

$$\rho = \frac{\sum_{i=1}^1 \sum_{j=1}^7 [(A_{ij} - \bar{A}) (B_{ij} - \bar{B})]}{\sqrt{[\sum_{i=1}^1 \sum_{j=1}^7 (A_{ij} - \bar{A})^2] [\sum_{i=1}^1 \sum_{j=1}^7 (B_{ij} - \bar{B})^2]}}$$

$$\bar{A} = \frac{1 + 2 + 9 + 9 + 8 + 2 + 1}{7} = 4.57$$

The first subwindow:

$$\bar{B}_1 = \frac{1 + 2 + 2 + 1 + 1 + 3 + 8}{7} = 2.57$$

$\rho_1 =$

$$\frac{(1-4.57)(1-2.57)+(2-4.57)(2-2.57)+(9-4.57)(2-2.57)+(9-4.57)(1-2.57)+(8-4.57)(1-2.57)+(2-4.57)(3-2.57)+(1-4.57)(8-2.57)}{\sqrt{[(1-4.57)^2+(2-4.57)^2+(9-4.57)^2+(9-4.57)^2+(8-4.57)^2+(2-4.57)^2+(1-4.57)^2] [(1-2.57)^2+(2-2.57)^2+(2-2.57)^2+(1-2.57)^2+(1-2.57)^2+(3-2.57)^2+(8-2.57)^2}} = -0.5$$

The second subwindow:

$$\bar{B}_2 = 3.57 , \text{ Correlation coefficient } \rho_2 = -0.59$$

The third subwindow:

$$\bar{B}_3 = 4.14 , \text{ Correlation coefficient } \rho_3 = -0.09$$

The fourth subwindow:

$$\bar{B}_4 = 4 , \text{ Correlation coefficient } \rho_4 = 0.69$$

The fifth subwindow:

$$\bar{B}_5 = 4 , \text{ Correlation coefficient } \rho_5 = 0.97$$

The sixth subwindow:

$$\bar{B}_6 = 4 , \text{ Correlation coefficient } \rho_6 = 0.37$$

Based on those results the fifth subwindow in the search window is the best match for the given template.

Example-6 (2-D correlation matching)

Given the following two subarrays from a pair of digital images. Compute the normalized correlation coefficient.

$$A = \begin{bmatrix} 48 & 50 & 48 \\ 51 & 67 & 71 \\ 55 & 67 & 72 \end{bmatrix}, \quad B = \begin{bmatrix} 52 & 37 & 41 \\ 67 & 54 & 78 \\ 42 & 74 & 87 \end{bmatrix}$$

Solution:

$$\rho = \frac{\sum_{i=1}^3 \sum_{j=1}^3 [(A_{ij} - \bar{A})(B_{ij} - \bar{B})]}{\sqrt{[\sum_{i=1}^3 \sum_{j=1}^3 (A_{ij} - \bar{A})^2] [\sum_{i=1}^3 \sum_{j=1}^3 (B_{ij} - \bar{B})^2]}}$$

$$\bar{A} = \frac{48 + 50 + 48 + 51 + 67 + 71 + 55 + 67 + 72}{9} = 58.5$$

$$\bar{B} = \frac{52 + 37 + 41 + 67 + 54 + 78 + 42 + 74 + 87}{9} = 59.1$$

$$\begin{aligned} \sum_{i=1}^3 \sum_{j=1}^3 [(A_{ij} - \bar{A})(B_{ij} - \bar{B})] &= \\ &= (48 - 58.8)(52 - 59.1) + (50 - 58.8)(37 - 59.1) \\ &+ (48 - 58.8)(41 - 59.1) + (51 - 58.8)(67 - 59.1) \\ &+ (67 - 58.8)(54 - 59.1) + (71 - 58.8)(78 - 59.1) \\ &+ (55 - 58.8)(42 - 59.1) + (67 - 58.8)(74 - 59.1) \\ &+ (72 - 58.8)(87 - 59.1) = 1149.22 \end{aligned}$$

$$\begin{aligned} \sum_{i=1}^3 \sum_{j=1}^3 (A_{ij} - \bar{A})^2 &= (48 - 58.8)^2 + (50 - 58.8)^2 + (48 - 58.8)^2 + (51 - 58.8)^2 + \\ &(67 - 58.8)^2 + (71 - 58.8)^2 + (55 - 58.8)^2 + (67 - 58.8)^2 + (72 - 58.8)^2 = 843.56 \end{aligned}$$

$$\sum_{i=1}^3 \sum_{j=1}^3 (B_{ij} - \bar{B})^2 = (52 - 59.1)^2 + (37 - 59.1)^2 + (41 - 59.1)^2 + (67 - 59.1)^2 + (54 - 59.1)^2 + (78 - 59.1)^2 + (42 - 59.1)^2 + (74 - 59.1)^2 + (87 - 59.1)^2 = 2604.89$$

$$\rho = \frac{1149.22}{\sqrt{843.56 \times 2604.89}} = 0.7753$$

Example-7

Given that T and S are template and search windows having two dimensional pattern. Determine normalized correlation coefficient between the template and all possible subwindows in the search window.

$$T = \begin{bmatrix} 30 & 50 & 30 \\ 50 & 50 & 50 \\ 30 & 50 & 30 \end{bmatrix} \qquad S = \begin{bmatrix} 44 & 45 & 64 & 50 & 45 & 43 \\ 43 & 44 & 63 & 48 & 49 & 45 \\ 45 & 47 & 65 & 50 & 45 & 45 \\ 62 & 62 & 69 & 64 & 64 & 63 \\ 48 & 48 & 68 & 51 & 55 & 50 \\ 41 & 44 & 63 & 48 & 42 & 47 \end{bmatrix}$$

Solution:

Correlation coefficients =	-0.3301	0.5405	-0.1745	-0.1217
	-0.3334	0.2394	-0.3921	-0.2718
	0.0585	0.9665	0.0806	0.6228
	-0.2321	0.2904	-0.1268	-0.1997

The maximum correlation coefficient occurs at 10th position whose central pixel is at c=3 and r=4 in the search window.

Example-8 (Subpixel position of 1-D correlation matching)

For the data of Example-5, compute

- A continuous correlation function (2nd order polynomial) in the neighborhood of the largest correlation coefficient.
- Location of the maximum of the formed correlation function.

Solution:

A = [1,2,9,9,8,2,1] , B = [1,2,2,1,1,3,8,8,6,1,1,1] , Best match B₅ = [1,3,8,8,6,1,1]

For subwindow B₄ (its center at column c =7), ρ =0.69

For subwindow B_5 (its center at column $c=8$), $\rho=0.97$

For subwindow B_6 (its center at column $c=9$), $\rho=0.37$

Equation of 2nd order polynomial: $\rho = a_0 + a_1 c + a_2 c^2$

Thus, three equations can be written for the three positions as follows

$$0.69 = a_0 + a_1 (7) + a_2 (7)^2$$

$$0.97 = a_0 + a_1 (8) + a_2 (8)^2$$

$$0.37 = a_0 + a_1 (9) + a_2 (9)^2$$

The equations can be expressed in a matrix form as follows

$$\begin{bmatrix} 1 & 7 & 49 \\ 1 & 8 & 64 \\ 1 & 9 & 81 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0.69 \\ 0.97 \\ 0.37 \end{bmatrix}$$

$$A \quad X = L$$

The solution can be given as

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = A^{-1} L = \begin{bmatrix} -25.91 \\ 6.88 \\ -0.44 \end{bmatrix}$$

To get Location of the maximum of the formed correlation function.

$$\frac{\partial \rho}{\partial c} = 0$$

$$a_1 + 2 a_2 c = 0$$

$$66.8 + (2)(-0.44) c = 0$$

$$c = 7.82$$

Example-9 (Subpixel position of 2-D correlation matching)

For the data of Example-7, show mathematically how to determine

- A continuous correlation function (2nd order 2D polynomial) in the neighborhood of the largest correlation coefficient.
- Location of the maximum of the formed correlation function.

Solution:

Equation of 2nd order 2D polynomial: $\rho = a_0 + a_1 c + a_2 r + a_3 c r + a_4 c^2 + a_5 r^2$

Since the pixel $P_{c,r}$ ($c=3,r=4$) of the largest correlation coefficient has 8 neighbors, 9 equations can be written for the pixel and its neighbors as follows:

$$\begin{aligned} \rho_{3-1,4-1} &= a_0 + a_1 (3-1) + a_2 (4-1) + a_3 (3-1) (4-1) + a_4 (3-1)^2 + a_5 (4-1)^2 \\ -0.3334 &= a_0 + a_1 (2) + a_2 (3) + a_3 (2) (3) + a_4 (2)^2 + a_5 (3)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3,4-1} &= a_0 + a_1 (3) + a_2 (4-1) + a_3 (3) (4-1) + a_4 (3)^2 + a_5 (4-1)^2 \\ 0.2394 &= a_0 + a_1 (3) + a_2 (3) + a_3 (3) (3) + a_4 (3)^2 + a_5 (3)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3+1,4+1} &= a_0 + a_1(3+1) + a_2(4-1) + a_3(3+1) (4-1) + a_4(3+1)^2 + a_5(4-1)^2 \\ -0.3921 &= a_0 + a_1 (4) + a_2(3) + a_3(4) (3) + a_4(4)^2 + a_5(3)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3-1,4} &= a_0 + a_1 (3-1) + a_2 (4) + a_3 (3-1) (4) + a_4 (3-1)^2 + a_5 (4)^2 \\ 0.0585 &= a_0 + a_1 (2) + a_2 (4) + a_3 (2) (4) + a_4 (2)^2 + a_5 (4)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3,4} &= a_0 + a_1 (3) + a_2 (4) + a_3 (3) (4) + a_4 (3)^2 + a_5 (4)^2 \\ 0.9665 &= a_0 + a_1 (3) + a_2 (4) + a_3 (3) (4) + a_4 (3)^2 + a_5 (4)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3+1,4} &= a_0 + a_1 (3+1) + a_2 (4) + a_3 (3+1) (4) + a_4 (3+1)^2 + a_5 (4)^2 \\ 0.0806 &= a_0 + a_1 (4) + a_2 (4) + a_3 (4) (4) + a_4 (4)^2 + a_5 (4)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3-1,4+1} &= a_0 + a_1(3-1) + a_2(4+1) + a_3(3-1) (4+1) + a_4(3-1)^2 + a_5(4+1)^2 \\ -0.2321 &= a_0 + a_1(2) + a_2(5) + a_3(2) (5) + a_4(2)^2 + a_5(5)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3,4+1} &= a_0 + a_1 (3) + a_2 (4+1) + a_3 (3) (4+1) + a_4 (3)^2 + a_5 (4+1)^2 \\ 0.2904 &= a_0 + a_1 (3) + a_2 (5) + a_3 (3) (5) + a_4 (3)^2 + a_5 (5)^2 \end{aligned}$$

$$\begin{aligned} \rho_{3+1,4+1} &= a_0 + a_1 (3+1) + a_2 (4+1) + a_3 (3+1) (4+1) + a_4 (3+1)^2 + a_5 (4+1)^2 \\ -0.1268 &= a_0 + a_1 (4) + a_2 (5) + a_3 (4) (5) + a_4 (4)^2 + a_5 (5)^2 \end{aligned}$$

3-1,4-1	3,4-1	3+1,4-1
3-1,4	3,4	3+1,4
3-1,4+1	3,4+1	3+1,4+1

By solving the 9 equations using L-S method the polynomial coefficients (a_0, a_1, \dots, a_5) can be determined.

$$\begin{aligned} a_0 &= -12.297 & a_1 &= 3.785 & a_2 &= 3.634 \\ a_3 &= 0.041 & a_4 &= -0.656 & a_5 &= -0.461 \end{aligned}$$

To get Location (c,r) of the maximum of the formed correlation function.

$$\frac{\partial \rho}{\partial c} = 0, \quad \frac{\partial \rho}{\partial r} = 0$$

which lead to

$$a_1 + a_3 r + 2 a_4 c = 0$$

$$a_2 + a_3 c + 2 a_5 r = 0$$

Solving those two equations together results in $c = 3.012$ and $r = 4.075$.

Least-squares matching

Different implementations of least-squares matching have been developed. The following mathematical formulation, which is commonly used, accounts for linear radiometric distortion and affine type (six parameters) geometric distortion.

$$G_1(x,y) = h_0 + h_1 G_2(x',y')$$

$$\text{in which } x' = a_0 + a_1x + a_2y \quad \text{and} \quad y' = b_0 + b_1x + b_2y$$

where $G_1(x, y)$ is the intensity value from the left image patch at position (x, y) , $G_2(x', y')$ is the intensity value from the right image patch at position (x', y') , h_0 and h_1 are radiometric transformation parameters, and a_0, a_1, a_2, b_0, b_1 and b_2 are affine transformation parameters relating corresponding pixel locations in the two images.

The last formulation can be expressed as a least-squares observation equation as follows:

$$G_1(x, y) + V_G(x, y) = h_0 + h_1 G_2(a_0 + a_1x + a_2y, b_0 + b_1x + b_2y)$$

$$\text{or } G = h_0 + h_1 G_2(a_0 + a_1x + a_2y, b_0 + b_1x + b_2y) = G_1(x, y) + V_G(x, y)$$

with G the function relating intensity values from the two patches, $V_G(x, y)$ the residual error and the other variables are as defined previously.

Using Taylor's theorem to linearize the last equation at initial approximations of the unknown radiometric and geometric parameters yields:

$$G = G_0 + dh_0 + G_2(x', y')dh_1 + h_1G_x da_0 + xh_1G_x da_1 + yh_1G_x da_2 \\ + h_1G_y db_0 + xh_1G_y db_1 + yh_1G_y db_2 = G_1(x, y) + V_G(x, y)$$

where G_x and G_y denote the partial derivatives of $G_2(x', y')$ with respect to x' and y' , respectively. Since G_2 includes digital numbers, G_x and G_y are found by estimating the slopes of G_2 in both the x and y directions using discrete values as follows:

$$G_x = [G_2(x'+1, y') - G_2(x'-1, y')]/2 \quad \text{and} \quad G_y = [G_2(x', y'+1) - G_2(x', y'-1)]/2$$

LS matching is an iterative process. This requires a fairly accurate initial value of the location of the right image patch. Also, initial approximations of the unknown radiometric and geometric parameters are needed. Initial estimates for h_0 and h_1 can be determined through the computation of the regression line that best fit the digital data of the two patches. The parameters a_0 and b_0 can be initially approximated by the differences between the center coordinates of the two patches, respectively. The other four parameters a_1 , a_2 , b_1 and b_2 are approximated by the values of 1, 0, 0 and 1, respectively, assuming initially that no rotation exists between the two images.

Before starting each iteration of the LS solution, the pixels of the right image patch G_2 , along with a one pixel wide border around the patch for derivative estimates, are resampled from the right image. This can be done by firstly transforming the x and y coordinates of each pixel of left image patch G_1 to the corresponding right image x', y' coordinates, using Equations $x' = a_0 + a_1x + a_2y$ and $y' = b_0 + b_1x + b_2y$, and updated parameters. Since x', y' coordinates would be usually fractional, the corresponding digital value is to be resampled utilizing digital values of neighboring pixels. Several resampling techniques are available and the most common ones are explained in the next Section.

Example-10 (1-D Least square matching)

Regarding Example-5, and after reaching the position of maximum normalized correlation coefficient, compute the optimum sub-pixel position of the template window with respect to the search window using the technique of least-squares (L-S) matching.

Solution:

$$T = [1, 2, 9, 9, 8, 2, 1] \quad , \quad \text{Best match in } S = [1, 3, 8, 8, 6, 1, 1]$$

L-S matching ignoring radiometric differences between the two windows

$$G_t(c) + V = G_s(c+dc)$$

$$G_t(c) + V = G_s(c) + \frac{\partial G_s(c)}{\partial c} dc$$

$$[G_t(c) - G_s(c)] + V = \frac{\partial G_s(c)}{\partial c} dc$$

$$\begin{bmatrix} 1 \\ 2 \\ 9 \\ 9 \\ 8 \\ 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 3 \\ 8 \\ 8 \\ 6 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{bmatrix} = \begin{bmatrix} (3-1)/2 \\ (8-1)/2 \\ (8-3)/2 \\ (6-8)/2 \\ (1-8)/2 \\ (1-6)/2 \\ (1-1)/2 \end{bmatrix} dc$$

$$dL + V = A dc$$

$$dc = (A^T A)^{-1} A^T dL = -0.2945$$

$$c = c^0 + dc = 8 + (-0.295) = 7.705$$

The solution is iterated until the increment dc approaches zero.

L-S matching considering radiometric differences between the two windows

$$G_t(c) + V = a G_s(c+dc) + b$$

$$G_t(c) + V = (a^0 G_s(c) + b^0) + \frac{a \partial G_s(c)}{\partial c} dc + G_s(c) da + db$$

Considering $a^0 = 1$ and $b^0 = 0$,

$$[G_t(c) - G_s(c)] + V = \frac{\partial G_s(c)}{\partial c} dc + G_s(c) da + db$$

$$\begin{bmatrix} 1 \\ 2 \\ 9 \\ 9 \\ 8 \\ 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 3 \\ 8 \\ 8 \\ 6 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 3.5 & 3 & 1 \\ 2.5 & 8 & 1 \\ -1 & 8 & 1 \\ -3.5 & 6 & 1 \\ -2.5 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} dc \\ da \\ db \end{bmatrix}$$

$$dL + V = A dX$$

$$dX = (A^T A)^{-1} A^T dL = \begin{bmatrix} -0.295 \\ 0.156 \\ -0.053 \end{bmatrix}$$

The values of parameters c, a, b are updated and the solution is iterated until the increments dc, da, db approach zero.

Example-11 (2-D Least square matching)

Regarding Example-7, and after reaching the position of maximum normalized correlation coefficient, compute the optimum sub-pixel position of the template window with respect to the search window using the technique of least-squares matching and considering radiometric differences between the two windows.

Solution:

$$T = \begin{bmatrix} 30 & 50 & 30 \\ 50 & 50 & 50 \\ 30 & 50 & 30 \end{bmatrix} \quad \text{Best match in } S = \begin{bmatrix} 47 & 65 & 50 \\ 62 & 69 & 64 \\ 48 & 68 & 51 \end{bmatrix}$$

L-S matching considering radiometric differences between the two windows

$$G_t(c,r) + V = a G_s(c+dc,r+dr) + b$$

$$G_t(c,r) + V = (a^0 G_s(c,r) + b^0) + \frac{a \partial G_s(c,r)}{\partial c} dc + \frac{a \partial G_s(c,r)}{\partial r} dr + G_s(c,r) da + db$$

Considering $a^0 = 1$ and $b^0 = -20$

$$[G_t(c,r) - (G_s(c,r) - 20)] + V = \frac{\partial G_s(c,r)}{\partial c} dc + \frac{\partial G_s(c,r)}{\partial r} dr + G_s(c,r) da + db$$

$$\begin{bmatrix} 30 \\ 50 \\ 30 \\ 50 \\ 50 \\ 50 \\ 30 \\ 50 \\ 30 \end{bmatrix} - \begin{bmatrix} 27 \\ 45 \\ 30 \\ 42 \\ 49 \\ 44 \\ 28 \\ 48 \\ 31 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \end{bmatrix} = \begin{bmatrix} (65 - 47)/2 & (62 - 47)/2 & 47 & 1 \\ (50 - 47)/2 & (69 - 65)/2 & 65 & 1 \\ (50 - 65)/2 & (64 - 50)/2 & 50 & 1 \\ (69 - 62)/2 & (48 - 47)/2 & 62 & 1 \\ (64 - 62)/2 & (68 - 65)/2 & 69 & 1 \\ (64 - 69)/2 & (51 - 50)/2 & 64 & 1 \\ (68 - 48)/2 & (48 - 62)/2 & 48 & 1 \\ (51 - 48)/2 & (68 - 69)/2 & 68 & 1 \\ (51 - 68)/2 & (51 - 64)/2 & 51 & 1 \end{bmatrix} \begin{bmatrix} dc \\ dr \\ da \\ db \end{bmatrix}$$

$$dL + V = A dX$$

$$dX = \begin{bmatrix} dc \\ dr \\ da \\ db \end{bmatrix} = (A^T A)^{-1} A^T dL = \begin{bmatrix} 0.194 \\ 0.086 \\ 0.138 \\ -5.372 \end{bmatrix}$$

The values of parameters c, r, a, b are updated and the solution is iterated until the increments dc, dr, da, db approach zero.

Digital Resampling

A digital image consists of a two-dimensional array of pixels. Each pixel is represented by a row index, a column index, and a digital intensity value indicating the lightness or darkness at its corresponding object location. Resampling is a process necessary to get a digital sample at a fractional row and column location. It involves interpolation between the intensity values of existing pixels to generate values that correspond to fractional locations. Several interpolation techniques are available for resampling digital images. However, the most common ones are known as nearest-neighbor interpolation, bilinear interpolation, and bicubic interpolation.

Nearest-Neighbor Interpolation

It is the simplest and the fastest of the three interpolation techniques. The interpolated intensity value is that of the image pixel whose center is closest to the fractional location. The fractional row and column indices are round off to the nearest integer values. The intensity value of the pixel at those integer indices is the desired interpolated value. For example the intensity value at the fractional location (c=543.33 and r=673.76) is the one existed at the integer location (c=543 and r=674).

Bilinear Interpolation

Greater computational complexity is involved in this technique than in nearest-neighbor interpolation. Here, the four pixels surrounding the fractional location are utilized in a linear interpolation in both column and row directions. Let that the considered pixel is located between the two columns of integer indices c and c+1 and the two rows of integer indices r and r+1. Let also dc and dr are the fractional row and column parts of the pixel, and that intensity value of at the position (c,r) is denoted by DN_{c,r}. For example if the fractional location is given by (c=543.33 and r=673.76), then c=543, c+1=544, r=673, r+1=674, dc=0.33 and dr=0.76.

	c	c+1
r	DN _{c,r}	DN _{c+1,r}
r+1	DN _{c,r+1}	DN _{c+1,r+1}

First, linearly interpolated values I_1 , I_2 are computed along rows IR and IR+1, respectively, as illustrated in the following two equations:

$$I_1 = DN_{c,r} + (dc)(DN_{c+1,r} - DN_{c,r})$$

$$I_2 = DN_{c,r+1} + (dc)(DN_{c+1,r+1} - DN_{c,r+1})$$

Next, a linear interpolation is done in the column direction, generating the final interpolated value I of the pixel. This is given in the following equation:

$$I = I_1 + (dr)(I_2 - I_1)$$

Bicubic Interpolation

It is the most rigorous resampling technique on the basis of signal processing theory. However, required computational time is considerable. Here, the sixteen pixels surrounding the fractional location are employed in an interpolation in both column and row directions by using a cubic spline function.

Example-12a

Referring to the 4x4 patch of a digital image given below, use the nearest neighbor and bilinear interpolation methods to get the interpolated value at the fractional location (c=547.44, r=817.24).

	c:545	c:546	c:547	c:548
r:817	68	64	72	79
r:818	64	71	78	74
r:819	61	65	71	78
r:820	63	56	62	68

Solution:

Nearest neighbor:

Fractional row and column indices are rounded to the nearest integer values and then intensity value of the pixel is determined at those integer indices.

For the fractional location $c=547.44$, $r=817.24$,

The nearest neighbor pixel is (547,817) and thus

$$DN_{547,817} = 72$$

Bilinear interpolation:

For the fractional location $c=547.44$, $r=817.24$,

The four surrounding pixels are considered, and interpolation along each of the two rows.

	c1: 547	c2: 548
r1: 817	72	79
r2: 818	78	74

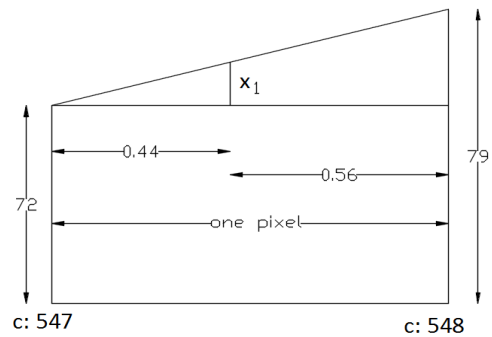
First row interpolation (817):

$$\frac{x_1}{(79-72)} = \frac{0.44}{1}$$

$$x_1 = 0.44 (79 - 72) = 3.08$$

Thus,

$$\text{The intensity value } I_1 = 72 + 3.08 = 75.08$$



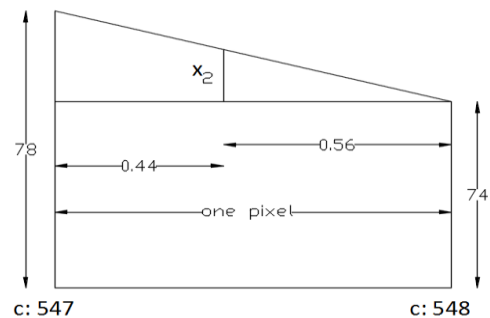
Second row interpolation (818):

$$\frac{x_2}{(78-74)} = \frac{0.56}{1}$$

$$x_2 = 0.56 (78 - 74) = 2.24$$

Thus,

$$\text{The intensity value } I_2 = 74 + 2.24 = 76.24$$

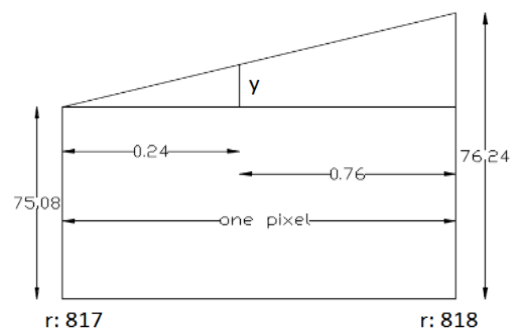


Interpolation in the column direction using the two previous interpolated values:

$$\frac{y}{(76.24-75.08)} = \frac{0.24}{1}$$

$$y = 0.24 (76.24 - 75.08) = 0.2784 ,$$

$$\text{The intensity value } I = 75.08 + 0.2784 = 75.35 \cong 75$$



Example-12b

Referring to the 4x4 patch of a digital image given below, use the nearest neighbor and bilinear interpolation methods to get the interpolated value at the fractional location (c=523.72m, r=835.24).

	c:523	c:524	c:525	c:526
r:835	58	54	66	64
r:836	54	62	67	59
r:837	52	57	60	54
r:838	53	46	52	48

Solution:

Nearest neighbor:

The fractional location c=523.72m, r=835.24

The nearest neighbor pixel is (524,835)

$$DN_{523.72, 835.24} = DN_{524, 835} = 54$$

Bilinear interpolation:

The fractional location c=523.72m, r=835.24

$$\begin{aligned} I_1 &= DN_{c,r} + (dc)(DN_{c+1,r} - DN_{c,r}) \\ &= 58 + 0.72(54-58) = 55.12 \end{aligned}$$

$$\begin{aligned} I_2 &= DN_{c,r+1} + (dc)(DN_{c+1,r+1} - DN_{c,r+1}) \\ &= 54 + 0.72(62-54) = 59.76 \end{aligned}$$

$$\begin{aligned} I &= I_1 + (dr)(I_2 - I_1) \\ &= 55.12 + 0.24(59.76 - 55.12) = 56.234 \cong 56 \end{aligned}$$

Interest point operators

Point features are usually referred to as interest points. Interest points are well-defined points or corners, found by using an interest point operator. Many point operators have been developed in the domains of computer vision and pattern recognition. Point operators don't rely only on spatial information but also apply statistics to determine how well-defined the position of a particular point is. Detected interest points in stereo scenes are often matched in order to create the shape of the imaged surface. Automated detection and matching of interest points play a major role in automating most photogrammetric processes, especially triangulation and generation of digital terrain model.

Here two well-known interest point operators are considered and explained. They are the Förstner operator and Moravec operator. Moravec's operator is the first interest point operator developed. It searches for points that have high variance between adjacent pixels. Forstner operator extract distinct points and corners by calculating the gradients and matrix N for each window within the image.

Förstner Operator

Förstner operator is applied on 5 by 5 or 7 by 7 windows through the image. It identifies distinct points using the matrix N as follows.

$$N = \begin{bmatrix} \sum(g'x)^2 & \sum g'x g'y \\ \sum g'x g'y & \sum(g'y)^2 \end{bmatrix}$$

Where:

$g'x$ [partial derivative at pixel (c,r) in the x or c (column) direction] = $[g(c+1, r) - g(c-1, r)]/2$

$g'y$ [partial derivative at pixel (c,r) in the y or r (row) direction] = $[g(c, r+1) - g(c, r-1)]/2$

$\sum(g'x)^2$ = Sum of Squares of $g'x$ at each pixel

$\sum(g'y)^2$ = Sum of Squares of $g'y$ at each pixel

$\sum g'x g'y$ = Sum of products of $g'x$ by $g'y$ at each pixel

The trace of N^{-1} is the measure of distinctness. The window is classified as an interest point if the trace exceeds certain threshold.

Example-13

Compute the N-matrix, related to application of Förstner interest operator to the following 5x5 window.

10	13	3	11	13
11	15	10	9	9
9	24	7	8	10
10	13	14	15	13
13	18	17	19	21

Solution:

The following table lists computed values necessary to evaluate the matrix N.

Pixel	g'_x	g'_y	$g'_x g'_y$	$(g'_x)^2$	$(g'_y)^2$
$P(c_2,r_2)=15$	$(10-11)/2=-0.5$	$(24-13)/2=5.5$	-2.75	0.25	30.25
$P(c_3,r_2)=10$	$(9-15)/2=-3$	$(7-3)/2=2$	-6	9	4
$P(c_4,r_2)=9$	$(9-10)/2=-0.5$	$(8-11)/2=-1.5$	0.75	0.25	2.25
$P(c_2,r_3)=24$	$(7-9)/2=-1$	$(13-15)/2=-1$	1	1	1
$P(c_3,r_3)=7$	$(8-24)/2=-8$	$(14-10)/2=2$	-16	64	4
$P(c_4,r_3)=8$	$(10-7)/2=1.5$	$(15-9)/2=3$	4.5	2.25	9
$P(c_2,r_4)=13$	$(14-10)/2=2$	$(18-24)/2=-3$	-6	4	9
$P(c_3,r_4)=14$	$(15-13)/2=1$	$(17-7)/2=5$	5	1	25
$P(c_4,r_4)=15$	$(13-14)/2=-0.5$	$(19-8)/2=5.5$	-2.75	0.25	30.25
sum	-9	17.5	-22.25	82	114.75

$$N = \begin{bmatrix} \sum(g'_x)^2 & \sum g'_x g'_y \\ \sum g'_x g'_y & \sum(g'_y)^2 \end{bmatrix}$$

$$N = \begin{bmatrix} 82 & -22.75 \\ -22.75 & 114.75 \end{bmatrix}$$

$$N^{-1} = \frac{1}{8891.9375} \begin{bmatrix} 114.75 & 22.75 \\ 22.75 & 82 \end{bmatrix}$$

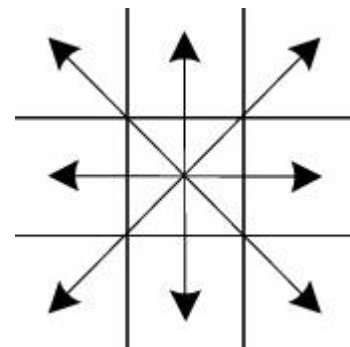
Trace of $N^{-1} = 0.02$

Moravec Operator

Moravec operator is applied on 4 by 4 or 8 by 8 windows through the image. It is implemented through the following procedure:

1. At each pixel at location (c,r) in the window, the differences between intensity value (DN) of surrounding pixels in the four directions (row, column and the two diagonals) are computed as follows:
 - Difference in the row direction = $DN_{c+1,r} - DN_{c-1,r}$
 - Difference in the column direction = $DN_{c,r+1} - DN_{c,r-1}$
 - Difference in the main diagonal direction = $DN_{c+1,r+1} - DN_{c-1,r-1}$
 - Difference in other diagonal direction = $DN_{c-1,r+1} - DN_{c+1,r-1}$

$DN_{c-1,r-1}$	$DN_{c,r-1}$	$DN_{c+1,r-1}$
$DN_{c-1,r}$	$DN_{c,r}$	$DN_{c+1,r}$
$DN_{c-1,r+1}$	$DN_{c,r+1}$	$DN_{c+1,r+1}$



2. The total sum of squared differences computed at each pixel is computed
3. The measure of distinctness is determined as the average of total sums of squared differences for all window pixels.
4. The window is classified as an interest point if that average exceeds certain threshold.

Example-14

Compute a measure of distinctness, related to application of Moravec interest operator to the following 4x4 window, considering only non-border pixels.

12	15	5	13
10	27	9	11
11	14	15	13
12	19	18	22

Solution:

Regarding $P(c_2, r_2)=27$,

$$\sum \text{differences}^2 = (9-10)^2 + (14-15)^2 + (11-5)^2 + (15-12)^2 = 47$$

Regarding $P(c_3, r_2)=9$,

$$\sum \text{differences}^2 = (11-27)^2 + (15-5)^2 + (14-13)^2 + (13-15)^2 = 361$$

Regarding $P(c_2, r_3)=14$,

$$\sum \text{differences}^2 = (11-15)^2 + (19-27)^2 + (12-9)^2 + (18-10)^2 = 153$$

Regarding $P(c_3, r_3)=15$,

$$\sum \text{differences}^2 = (13-14)^2 + (18-9)^2 + (19-11)^2 + (22-27)^2 = 171$$

$$\text{Mean} = \frac{47+361+153+171}{4} = 183$$

Smoothing Filters

The role of filters is to filter unwanted signals in the image. Averaging and weighted Averaging filters are two examples of smoothing (low-pass) filters that reduce the image noise.

- 3×3 averaging filter takes the form

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

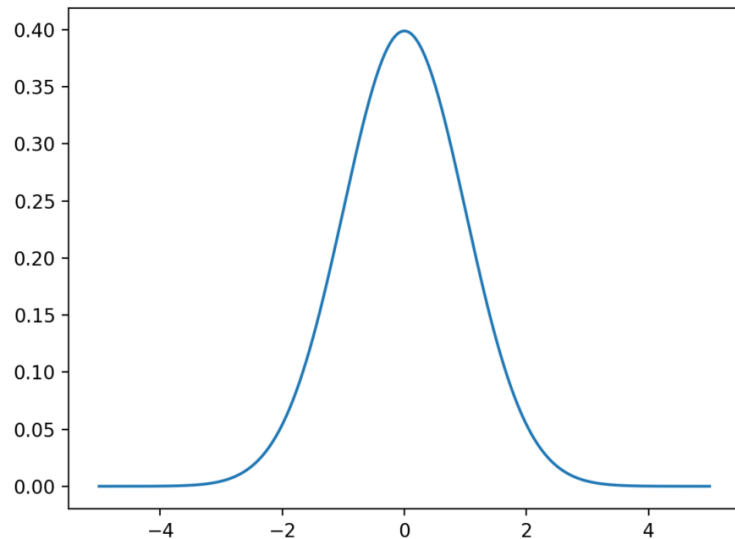
- 3×3 weighted averaging filter assign more weight for the central pixel. One example is shown below.

	1	1	1
1/10	1	2	1
	1	1	1

Gaussian filter is another smoothing filter that is sampled from the 2-D Gaussian function. The figure below shows 1-D Gaussian function with $\sigma= 1$.

The equation of 1-D Gaussian function:

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



The equation of a 2-D Gaussian function can be written as

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

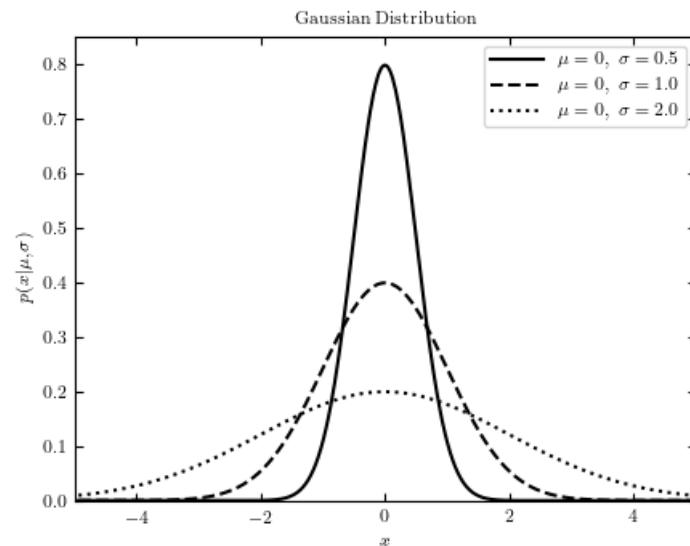
The 3 by 3 kernel shown below is an approximation of 2-D Gaussian filter with $\sigma= 1$

	1	2	1
1/16	2	4	2
	1	2	1

The 5 by 5 kernel shown below is an approximation of 2-D Gaussian filter with $\sigma=1$

	1	4	6	4	1
	4	16	24	16	4
1/256	6	24	36	24	6
	4	16	24	16	4
	1	4	6	4	1

As illustrated in the figure to the right, Larger values of σ yield wider peaks and greater blurring. Kernel size must increase with larger values of σ to keep the Gaussian nature of the filter. In addition, the values of Kernel coefficients differ according to the value of σ .

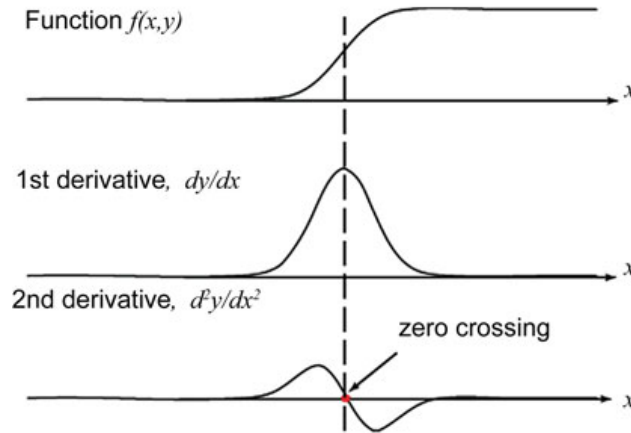


Edge detectors

Edges represent significant changes in intensity values in the image. Edge detection is the process of locating the edge pixels. If the edges in an image can be identified accurately, image objects can be located, and their basic properties such as area, perimeter, and shape can be measured. Since image analysis involves the identification and classification of objects in an image, edge detection is an essential tool.

There are two groups of edge detectors:

- Detectors that approximate first derivative.
- Detectors based on the zero-crossings of the image function second derivative.



Prewitt, Sobel operators are examples of first derivative operators:

- Prewitt edge detector (in row and column directions)

$$pk_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$pk_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Edge magnitude = $(Gx^2 + Gy^2)^{1/2}$ where Gx and Gy are the responses to pk_x and pk_y , respectively.

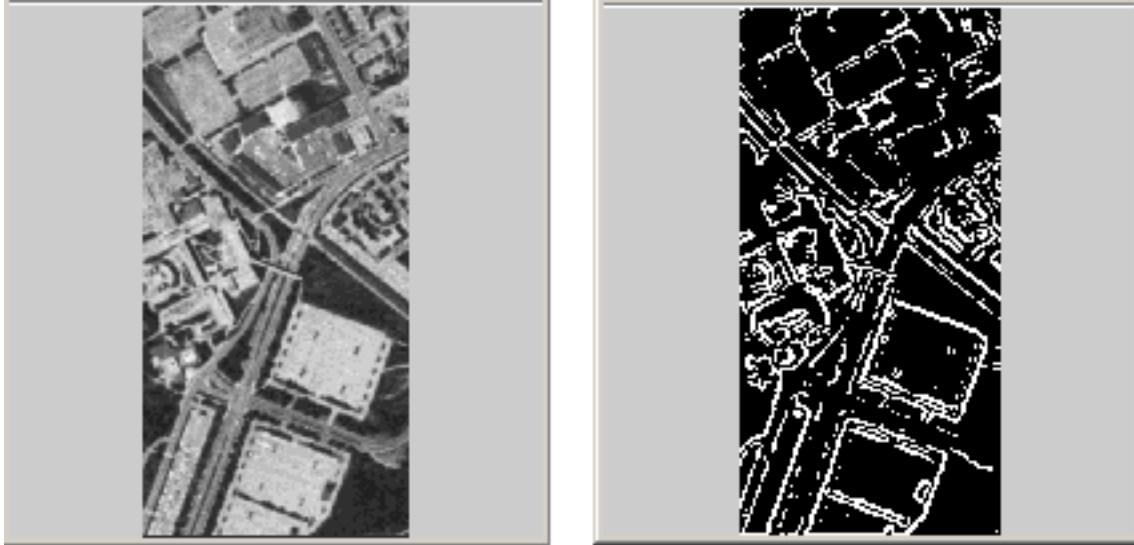
- Sobel edge detector (in row and column directions)

$$sk_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$sk_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Edge magnitude = $(Gx^2 + Gy^2)^{1/2}$ where Gx and Gy are the responses to sk_x and sk_y , respectively.

The figure below shows an image patch and related edges generated by Sobel operator.



Laplacian operator is an example of second derivative operators. It has the two following forms

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

Note that the eight pixels surrounding the first form of Laplacian operator (to the left) have different coefficients (zeros and ones) where the second filters have adjacent pixels of equal coefficients (ones).

Laplacian of Gaussian (LOG) is a combination of the Laplacian and Gaussian operators where the image is smoothed first by Gaussian operator to suppress the noise in the image before applying Laplacian operator for edge detection.

Example-15

Convolve the window below with the each of the following 3×3 filters/edge operators:

- Averaging filter.
- Gaussian filter.
- Sobel operator (in row and column directions).

- Prewitt operator (in row and column directions).

10	13	3	11	13
11	15	10	9	9
9	24	7	8	10
10	13	14	15	13
13	18	17	19	21

Also, determine the edge magnitude at each pixel resulted by applying the last two operators.

Solution:

- Averaging filter:

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

- For pixel $(r_2, c_2)=15$, result $=1/9(10+13+3+11+15+10+9+24+7)=11.33$
- For pixel $(r_2, c_3)=10$, result $=1/9(13+3+11+15+10+9+24+7+8)=11.11$
- For pixel $(r_2, c_4)=9$, result $=1/9(3+11+13+9+9+10+7+8+10)=8.89$
- For pixel $(r_3, c_2)=24$, result $=1/9(11+15+10+9+24+7+10+13+14)=12.55$
- For pixel $(r_3, c_3)=7$, result $=1/9(15+10+9+24+7+8+13+14+15)=12.78$
- For pixel $(r_3, c_4)=8$, result $=1/9(10+9+9+7+8+10+14+15+13)=10.55$
- For pixel $(r_4, c_2)=13$, result $=1/9(9+24+7+1+13+14+13+18+17)=12.88$
- For pixel $(r_4, c_3)=14$, result $=1/9(24+7+8+13+14+15+18+17+19)=15$
- For pixel $(r_4, c_4)=15$, result $=1/9(7+8+10+14+15+13+17+19+21)=13.78$

Resulted convolved image:

11.3	11.1	8.9
12.5	12.8	10.5
12.9	15.0	13.8

- Gaussian filter:

	1	2	1
1/16	2	4	2
	1	2	1

- For pixel $(r_2, c_2)=15$, result $=1/16(10+2 \times 13+3+11 \times 2+15 \times 4+2 \times 10+9+2 \times 24+7)=12.81$
- For pixel $(r_2, c_3)=10$, result $=1/16(13+2 \times 3+11+2 \times 15+4 \times 10+2 \times 9+24+2 \times 7+8)=10.25$
- For pixel $(r_2, c_4)=9$, result $=1/16(3+2 \times 11+13+9 \times 4+2 \times 9+2 \times 10+7+8 \times 2+10)=9.1$
- For pixel $(r_3, c_2)=24$, result $=1/16(11+2 \times 15+10+9 \times 2+4 \times 24+7 \times 2+10+2 \times 13+14)=14.3$
- For pixel $(r_3, c_3)=7$, result $=1/16(15+2 \times 10+9+24 \times 2+7 \times 4+8 \times 2+13+14 \times 2+15)=12$
- For pixel $(r_3, c_4)=8$, result $=1/16(10+2 \times 9+9+7 \times 2+8 \times 4+10 \times 2+14+15 \times 2+13)=10$
- For pixel $(r_4, c_2)=13$, result $=1/16(9+2 \times 24+7+2 \times 10+4 \times 13+2 \times 14+13+18 \times 2+17)=14.4$
- For pixel $(r_4, c_3)=14$, result $=1/16(24+7 \times 2+8+13 \times 2+14 \times 4+15 \times 2+18+17 \times 2+19)=14.3$
- For pixel $(r_4, c_4)=15$, result $=1/16(7+8 \times 2+10 \times 1+14 \times 2+15 \times 4+13 \times 2+17+19 \times 2+21)=13.94$

Convolved image:

12.8	10.2	9.1
14.3	12.0	10.0
14.4	14.3	13.9

- Prewitt edge detector (in row and column directions)

$$pk_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$pk_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- For pixel $(r_2, c_2)=15$: $g_x=10+13+3-9-24-7=-14$ and $g_y=-10+3-11+10-9+7=-10$
- For pixel $(r_2, c_3)=10$: $g_x=13+3+11-24-7-8=-12$ and $g_y=-13-15-24+11+9+8=-24$
- For pixel $(r_2, c_4)=9$: $g_x=3+11+13-7-8-10=2$ and $g_y=-3-10-7+13+9+10=12$
- For pixel $(r_3, c_2)=24$: $g_x=11+15+10-10-13-14=-1$ and $g_y=-11-9-10+10+7+14=1$
- For pixel $(r_3, c_3)=7$: $g_x=15+10+9-13-14-15=-8$ and $g_y=-15-24-13+9+8+15=-20$
- For pixel $(r_3, c_4)=8$: $g_x=10+9+9-14-15-13=-14$ and $g_y=-10-7-14+9+10+13=1$
- For pixel $(r_4, c_2)=13$: $g_x=9+24+7-13-18-17=-8$ and $g_y=-9-10-13+7+14+17=6$
- For pixel $(r_4, c_3)=14$: $g_x=24+7+8-18-17-19=-15$ and $g_y=-24-13-18+8+15+19=-13$
- For pixel $(r_4, c_4)=15$: $g_x=7+8+10-17-19-21=-32$ and $g_y=-7-14-17+10+13+21=6$

$$G_x = \begin{bmatrix} -14 & -12 & 2 \\ -1 & -8 & -14 \\ -8 & -15 & -32 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -10 & -24 & 12 \\ 1 & -20 & 1 \\ 6 & -13 & 6 \end{bmatrix}$$

$$\text{Edge magnitude} = \begin{bmatrix} 17.2 & 26.8 & 12.2 \\ 1.4 & 21.54 & 14.0 \\ 10.0 & 19.85 & 32.6 \end{bmatrix}$$

- Sobel edge detector (in row and column directions)

$$sk_x = \begin{matrix} & \begin{matrix} 1 & 2 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & & \\ \begin{matrix} -1 \\ -2 \\ -1 \end{matrix} & & \end{matrix}$$

$$sk_y = \begin{matrix} \begin{matrix} -1 \\ -2 \\ -1 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 1 \end{matrix} \end{matrix}$$

- For pixel $(r_2, c_2)=15$: $g_x=10+2 \times 13+3-9-2 \times 24-7=-25$ and $g_y=-10-2 \times 11-9+3+2 \times 10+7=-11$
- For pixel $(r_2, c_3)=10$: $g_x=13+2 \times 3+11-24-2 \times 7-8=-16$ and $g_y=-13-2 \times 15-24+11+2 \times 9+8=-30$
- For pixel $(r_2, c_4)=9$: $g_x=3+2 \times 11+13-7-2 \times 8-10=5$ and $g_y=-3-2 \times 10-7+13+2 \times 9+10=11$
- For pixel $(r_3, c_2)=24$: $g_x=11+2 \times 15+10-10-2 \times 13-14=1$ and $g_y=-11-2 \times 9-10+10+2 \times 7+14=-1$
- For pixel $(r_3, c_3)=7$: $g_x=15+2 \times 10+9-13-2 \times 14-15=-12$ and $g_y=-15-2 \times 24-13+9+2 \times 8+15=-36$
- For pixel $(r_3, c_4)=8$: $g_x=10+2 \times 9+9-14-2 \times 15-13=-20$ and $g_y=-10-2 \times 7-14+9+2 \times 10+13=4$
- For pixel $(r_4, c_2)=13$: $g_x=9+2 \times 24+7-13-2 \times 18-17=-2$ and $g_y=-9-2 \times 10-13+7+2 \times 14+17=10$
- For pixel $(r_4, c_3)=14$: $g_x=24+2 \times 7+8-18-2 \times 17-19=-25$ and $g_y=-24-2 \times 13-18+8+2 \times 15+19=-11$
- For pixel $(r_4, c_4)=15$: $g_x=7+2 \times 8+10-17-2 \times 19-21=-43$ and $g_y=-7-2 \times 14-17+10+2 \times 13+21=5$

$$G_x = \begin{matrix} \begin{matrix} -25 & -16 & 5 \end{matrix} \\ \begin{matrix} 1 & -12 & -20 \end{matrix} \\ \begin{matrix} -2 & -25 & -43 \end{matrix} \end{matrix}$$

$$G_y = \begin{matrix} \begin{matrix} -11 & -30 & 11 \end{matrix} \\ \begin{matrix} -1 & -36 & 4 \end{matrix} \\ \begin{matrix} 10 & -11 & 5 \end{matrix} \end{matrix}$$

$$\text{Edge magnitude} = \begin{matrix} \begin{matrix} 27.3 & 34.0 & 12.1 \end{matrix} \\ \begin{matrix} 1.4 & 37.9 & 20.4 \end{matrix} \\ \begin{matrix} 10.2 & 27.3 & 43.3 \end{matrix} \end{matrix}$$

Example-16

Convolve the following 5×5 window

1	2	1	1	1
1	1	9	2	1
1	9	9	8	1
1	1	8	1	2
2	1	1	1	1

With the Laplacian operator given by

1	1	1
1	-8	1
1	1	1

Solution:

- For pixel $(r_2, c_2)=1$: result = $1+2+1+1-8+9+1+9+9=25$
- For pixel $(r_2, c_3)=9$: result = $2+1+1+1-9 \times 8+2+9+9+8=-39$
- For pixel $(r_2, c_4)=2$: result = $1+1+1+9-2 \times 8+1+9+8+1=15$
- For pixel $(r_3, c_2)=9$: result = $1+1+9+1-8 \times 9+9+1+1+8=-41$
- For pixel $(r_3, c_3)=9$: result = $1+9+2+9-9 \times 8+8+1+8+1=-33$
- For pixel $(r_3, c_4)=8$: result = $9+2+1+9-8 \times 8+1+8+1+2=-31$
- For pixel $(r_4, c_2)=1$: result = $1+9+9+1-8+8+2+1+1=24$
- For pixel $(r_4, c_3)=8$: result = $9+9+8+1-8 \times 8+1+1+1+1=-33$
- For pixel $(r_4, c_4)=1$: result = $9+8+1+8-8+2+1+1+1=23$

Convolved Image:

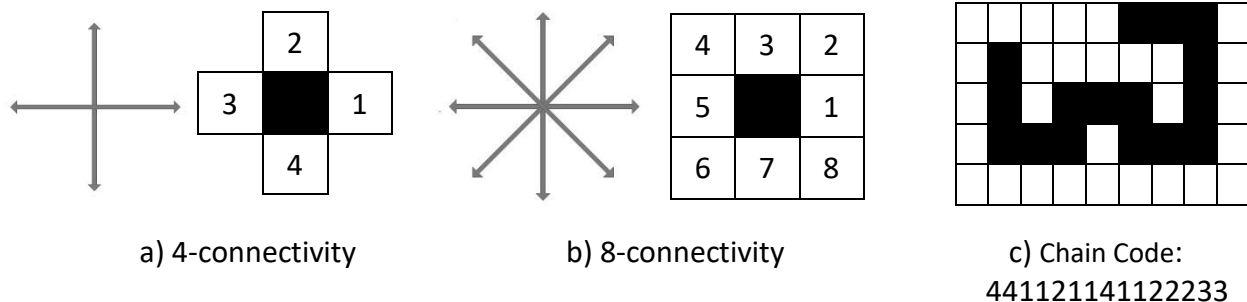
25	-39	15
-41	-33	-31
24	-33	23

Digital edge representation and description

Digital edges usually represent boundaries of objects in digital images. Object boundary is a connected set of points. Having a binary image, object boundary points are represented by ones (1's) and points not on the boundary, named background points, are represented by zeros (0's). This boundary data is to be compacted into representation that is more useful in computation of boundary features such as its length and the number of concavities it contains. Chain codes, minimum-perimeter polygon, signatures and boundary segments can be used for object boundary representation. On the other hand, boundary descriptors are used to describe object boundary. Simple descriptors, shape numbers and statistical moments are examples to boundary descriptors. Descriptors should be insensitive as possible to object size (scale), rotation and translation. However, descriptors mentioned above satisfy one or more of these properties.

Chain code (Freeman code)

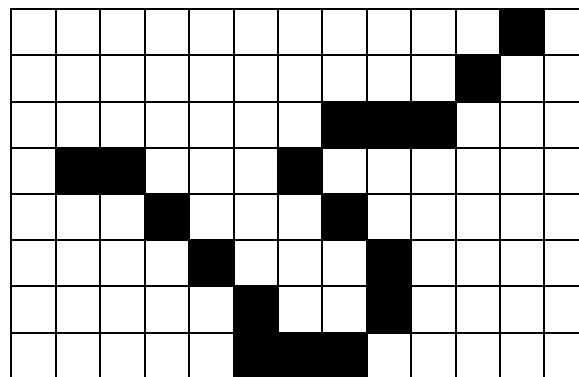
Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. This representation is based on 4- or 8-connectivity of the segments. The direction of each segment is coded by a numbering scheme such as the ones shown in the figure below (Figures a and b). Chain codes based on this scheme are referred to as Freeman chain code. Figure c displays the chain code based on 4-connectivity for a sample digital edge.



The chain code of a boundary is sensitive to image rotation. To overcome this problem, differential chain code, which is invariant to rotation, can be employed instead. Its elements would be computed as the differences among the chain code elements. Having a closed boundary, the first element of the differential chain code is computed as the difference between the last and first elements of the chain code.

Example-17

Determine the chain code and differential chain code for the digital edge shown below, starting from the left end of the edge.



Solution:

Chain code:

Starting from the left end of the edge the following code will be resulted:

1 8 8 8 7 1 1 2 3 4 4 2 1 1 2 2

4	3	2
5		1
6	7	8

Differential chain code:

By subtracting chain code numbers successively, we obtain the following number:

7 0 0 -1 -6 0 1 1 1 0 -2 -1 0 1 0

The negative numbers are converted to positive numbers by adding 8 (the total numbers of directions) to each of them leading to the following differential chain code:

7 0 0 7 2 0 1 1 1 0 6 7 0 1 0

Exercise

Repeat solving the example above but here it is required to start coding from the right end of the given edge.

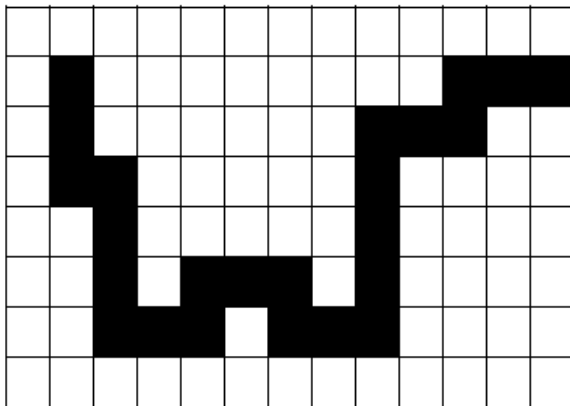
Some Simple boundary descriptors

1- The length of a boundary

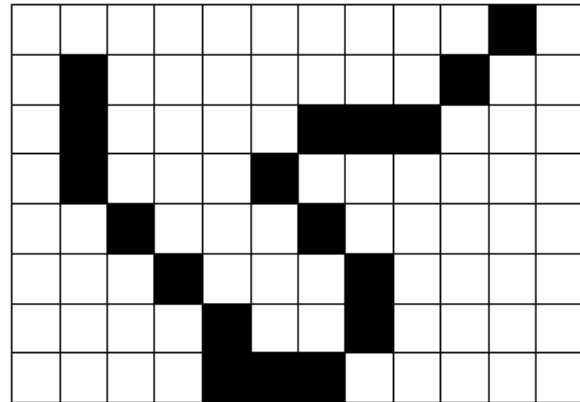
The length of a 4-connected boundary is simply the number of pixels in the boundary minus 1. If the boundary is 8-connected, the vertical and horizontal transitions are counted as 1 and the diagonal transitions as $\sqrt{2}$.

Example-18

Determine the length of digital edges shown below.



(a)



(b)

Solution:

For the curve (a) to the left, No. of curve points = 24
 Thus, curve length in pixel units = 24-1 = 23

Regarding the curve (b) to the right, No. of vertical transitions= 4, No. of horizontal transitions=4,
 No. of diagonal transitions = 9. Thus, curve length in pixel units = 4 + 4 + 9 $\sqrt{2}$ = 20.7

2- The diameter of a boundary

The diameter of a boundary is defined as the distance between the two farthest points on the boundary. The line segments connecting these points is called the major axis of the boundary. The minor axis of the boundary would be perpendicular to the major axis and of such length that the box passing through the outer four points of intersection of the boundary with the two axes. This box is called the basic rectangle, and the ratio of the major axis to the minor axis is called the eccentricity of the boundary.

