Prof.Dr.Azmi Tawfiq. Hussain Al-Rawi,
University of Anbar of Commuter science & I.T.
Department of Computer science \4th stage
Image processing

Digital image processing

(Introduction & Fundamentals)

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solution to a given problem. Digital image processing is concerned with computer processing of pictures, or more generally, images that have been converted into a numeric form.

Image sources: The basic function of an image source is to represent the distribution of light energy in an image as an electronic signal which may be communicated to a computer for storage and subsequent analysis or enhancement. These images that have been sources, for example digital cameras, scanner, satellite sensors, Sources may be classified as:

- Area scan or line scan. The linear array itself may be progressively moved across a static object as in the case of document scanner.
- Standard scan formats from broadcast television system or non-standard scan format include moonshot operation and multi-frame scanning
- Visible sources are covered on cameras, or non-visible spectral responsive to infrared-red, ultra-violet, X-ray radiation

Picture &image:

Picture: An individual painting, drawing or other representation on a surface of an objects such a representation as a work of art.

Image: A digital representation of an optical scene logos or graphs.

Image representation:

*Image usually represented as a matrix of cells called pixel.

*Every pixel in image has the same particular strong size.

*Depending on pixel's strong size images can be classified.

*Spectral resolution also depends on size of a pixel.

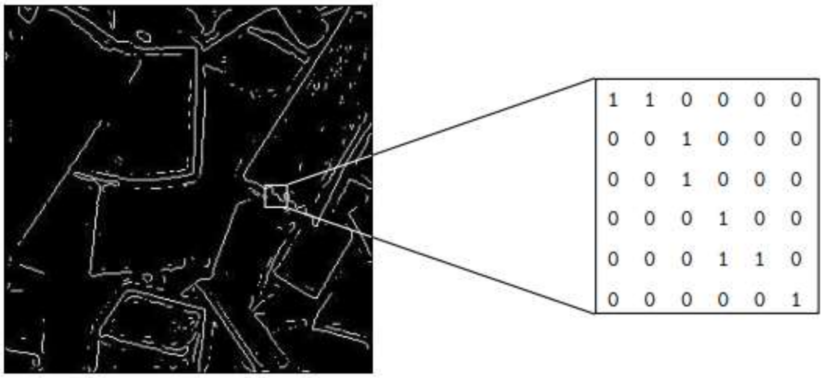
Types of image (Image classified):

A-Single band image:

* Binary image (1 bit for one pixel) (color value 0 or 1). See Fig. (1)

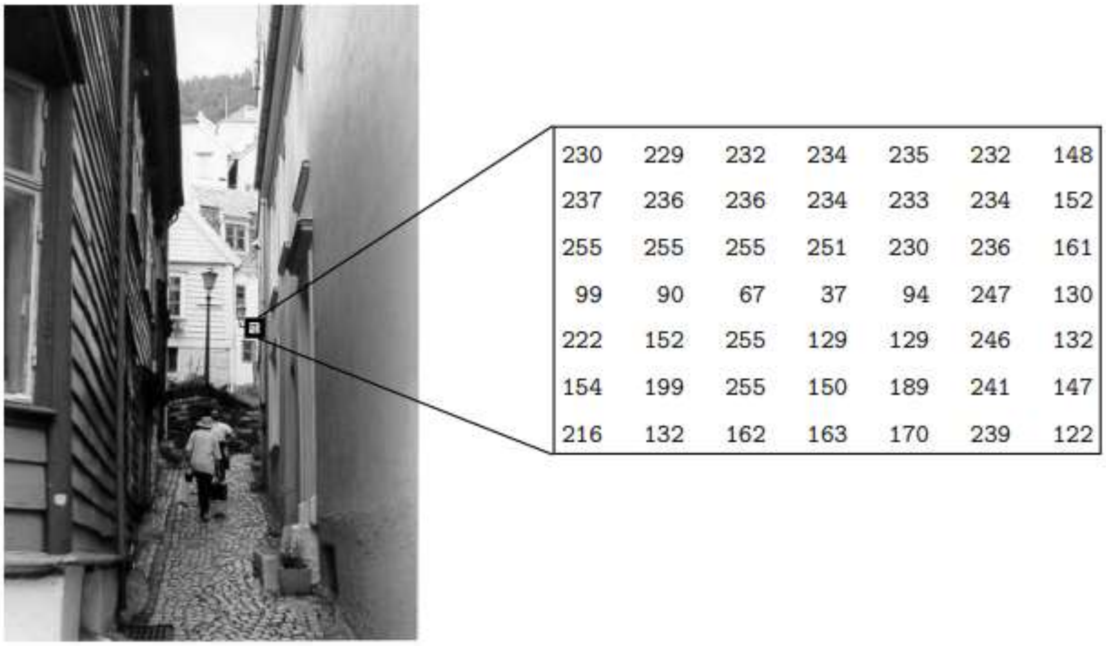
*Intensity image pixel (8 bits for one pixel) (color value 0-255). See Fig.(2)

*Intensity image pixel can be more than 8 bits for example 16 or 32 bits, and that depends on spectral resolution.



1	1	0	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	0	0	0	1

Figure (1): Binary Image.



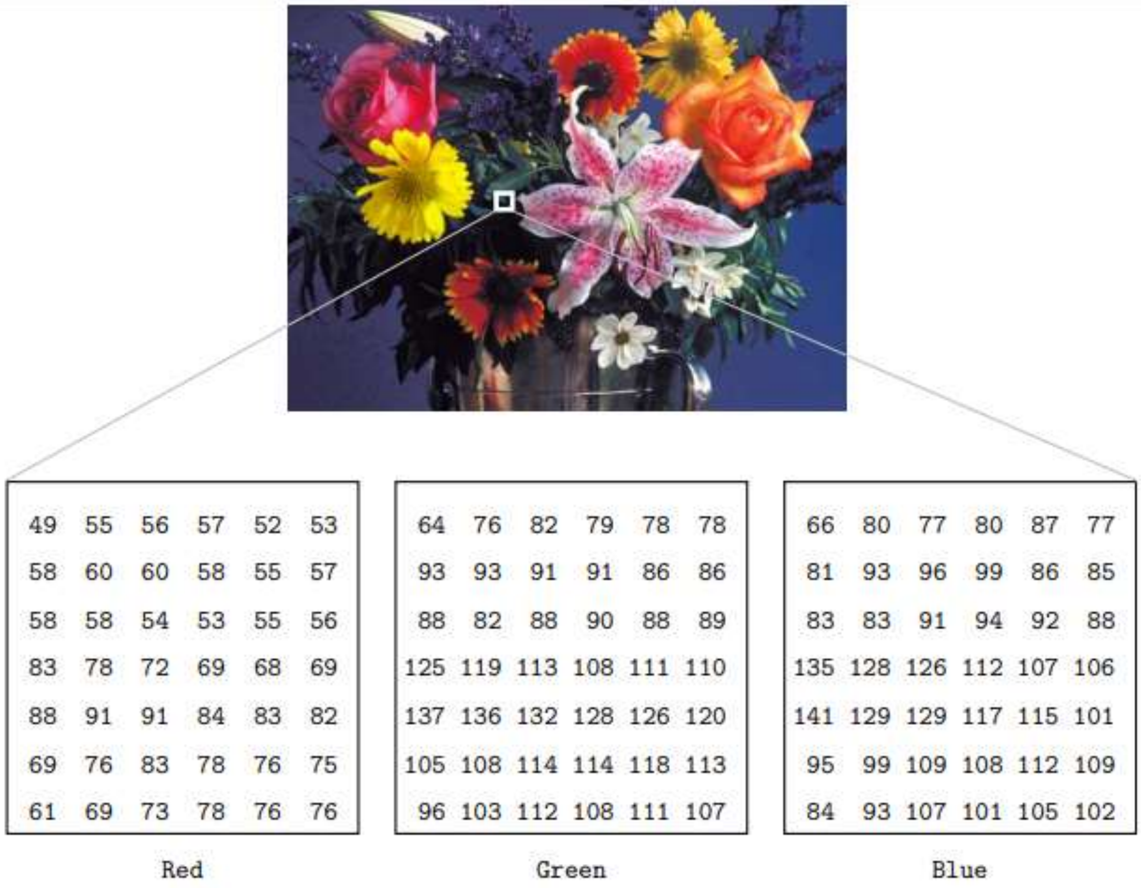
230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Figure (2): A greyscale image

B-Multi band images:

*Three bands (red, green, and blue(RGB)) (3x8 bits for one pixel) (color value 0-255 for each band of pixel). This gives a total of 256^3 =16777216 different possible colors in the image. See Fig.(3)

*Number of bands can be more than three such as data of the USGS (the united states geological survey) for example land sat which has 8 bands.



49	55	56	57	52	53
58	60	60	58	55	57
58	58	54	53	55	56
83	78	72	69	68	69
88	91	91	84	83	82
69	76	83	78	76	75
61	69	73	78	76	76

Red

64	76	82	79	78	78
93	93	91	91	86	86
88	82	88	90	88	89
125	119	113	108	111	110
137	136	132	128	126	120
105	108	114	114	118	113
96	103	112	108	111	107

Green

66	80	77	80	87	77
81	93	96	99	86	85
83	83	91	94	92	88
135	128	126	112	107	106
141	129	129	117	115	101
95	99	109	108	112	109
84	93	107	101	105	102

Blue

Figure (3): A true color image

Digital Image Basics:

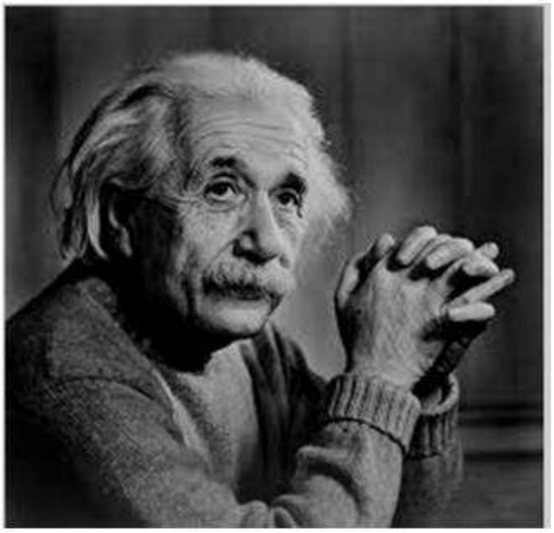
- An image may be define as a two dimensional function f(x,y), where x and y are spatial coordinates and the amplitude at any pair of coordinates (x,y) is called **intensity** or **gray**

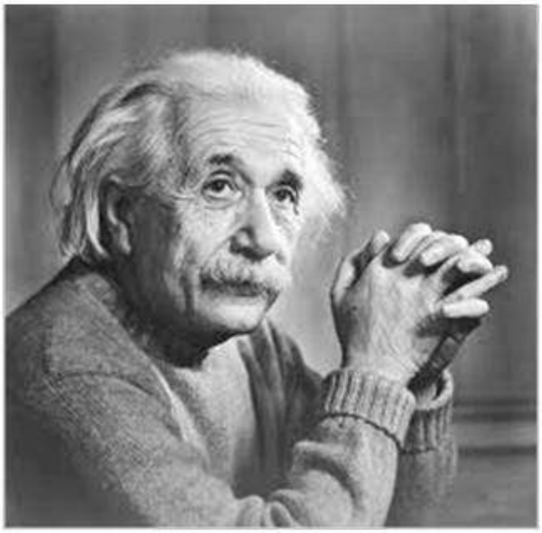
level of the image at that point. When x, y and the amplitude values of all are finite discrete quantities, we call the image digital image.

- A digital image is composed of a finite number of elements each of which has a particular location or value. These elements are referred to as pixels or image elements or picture elements or pixels elements.
- Gray level refers to a scalar measure of intensity that ranges from black to grays and finally to white or refer to the intensity of monochromic images.
- Color images are formed by a combinations of individual 2-D images. For examples, in the RGB color system, a color image consists of three individual component images.
- An image may be continuous with respect to the x and y –coordinates and also in amplitude. See Figure (4).

Figure (4): Continuously varying intensities

- Converting such an image to digital form requires that the coordinates as well as the amplitude be digitizes.
- Digitization the coordinate values is called **sampling**; Digitization the amplitude values is called **quantization**.
- The range of values spanned by the gray scale is called **dynamic range of an image**. Image will have high contrast, if the dynamic range is high and image will have dull washed out gray look if the dynamic range is low.
- The spatial coordinates of a digital image (x,y) are proportional to brightness. Brightness is a relative term. It depends on your visual perception. Since brightness is a relative term, so brightness can be defined as the amount of energy output by a source of light relative to the source we are comparing it to. In some cases, we can easily say that the image is bright, and in some cases, it's not easy to perceive





- We can easily see, that the image on the right side is brighter as compared to the image on the left.
- But if the image on the right is made more darker then the first one, then we can say that the image on the left is more brighter then the left.
- in DIP “intensity” is a numerical value which represents a pixel, i.e. for a 8-bit gray scale image the value of a pixel can be in the range of [0,255], where (generally) numerical “0” represents black color and numerical “255” represents white color. Therefore, the higher the value of a pixel, i.e. intensity, the pixel will appear to be whiter.
- Whereas, “brightness” is a relative term which can be understood visually. If two pixels (say, pixel1 and pixel2) of an image have intensity values, 125 and 210 respectively, then pixel2 is brighter than pixel1.
- Brightness of an object is the perceived luminance of the surround.
- Two objects with different surroundings would have identical luminance but different brightness.
- Luminance measured in lumens (lm), gives a measure of the amount of energy an observer perceiver from a light source.
- **Illumination** is the amount of source light incident on the scene. It is represented as i(x, y). **Reflectance** is the amount of light reflected by the object in the scene. It is represented by r(x, y).
- **Subjective brightness** means intensity as preserved by the human visual system. **Brightness adaptation** means the human visual system can operate only from scotopic to glare limit. It cannot operate over the range simultaneously. It accomplishes this large variation by changes in its overall intensity.
- Machband effect means the intensity of the stripes is constant. Therefore it preserves the brightness pattern near the boundaries, these bands are called as **Machband effect**.
- **Shrinking of image** may be viewed as under sampling. To shrink an image by one half, we delete every row and column. To reduce possible aliasing effect, it is a good idea to blue an image slightly before shrinking it.

- **Hue** is a color attribute that describes a pure color where **saturation** gives a measure of the degree to which a pure color is diluted by white light.
- **Resolution** is defined as the smallest number of discernible detail in an image. Spatial resolution is the smallest discernible detail in an image and gray level resolution refers to the smallest discernible change is gray level.
- Every pixel has number of bits (k)
- The number of bits required to store a digital image is:

$$S=N \times M \times k$$

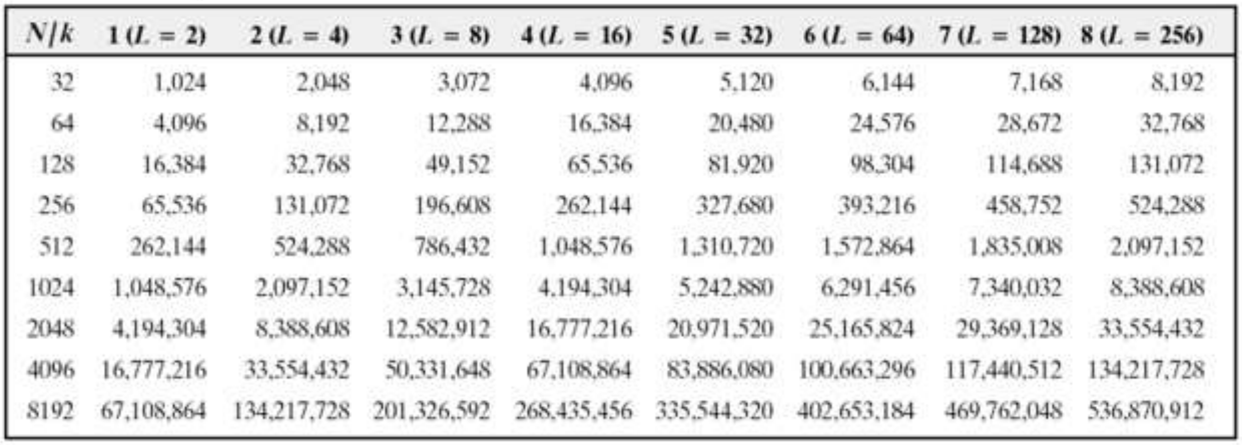
NxM: Number of pixels

L is max of gray levels. the number of gray levels typically is an integer power of 2 i.e. $L = 2^k$. See table (1).

- Measure unit in digital image Ppi or Dpi i.e the number of pixels in determine area (one inch).
- Image resolution =(150-300)Ppi in design print of paper.
- Image resolution =69Ppi to store the image and display on the computer screen.
- Image resolution =72Ppi to post the image in internet.
- Convert the image resolution from pixels to inch:
- Length the image in inch=image length in pixel/Ppi
- width the image in inch=image width in pixel/Ppi
- Zooming may be viewed as over sampling. It involves the creation of new pixel locations and the assignment of gray levels to those new locations.
- The real image compressed to keep memory before it be stored using algorithms, that depend on encoding technique (PCX, Bmp, TGA , TIFF, JPG,….).

Table (1)

Number of storage bits for various values of *N* and *k*.



N/k	$1\ (L = 2)$	$2\ (L = 4)$	$3\ (L = 8)$	$4\ (L = 16)$	$5\ (L = 32)$	$6\ (L = 64)$	$7\ (L = 128)$	$8\ (L = 256)$
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

Coordinate Convention

The result of sampling and quantization is a matrix of real number. See figure 1. That means: Digitization the coordinate values is called **sampling**; Digitization the amplitude values is called **quantization**

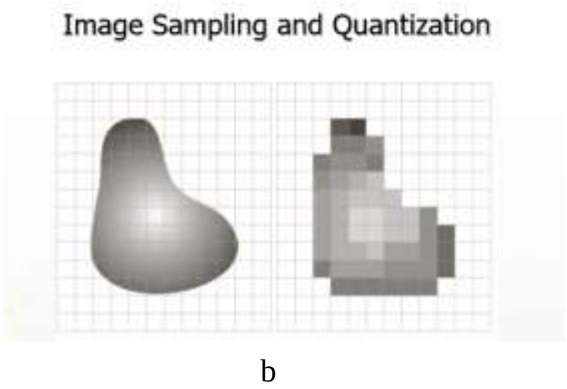


Fig.(1): a) Continuous image projected onto a sensor array b) Result of image quantization and sampling.

Assume that an image f(x,y) is a sampled has M rows and N columns. We say the image size MxN. The values of the coordinates (x,y) are discrete quantities.

The image origin is define to be at (x,y)=(0,0) . The next coordinate values of the first row of the image (x,y)=(0,1); thus the range of x from 0 to M-1, and the range of y from 0 to N-1.

If (r,c)=(1,1); thus the range from 1 to M, and from 1 to N. Coordinate conventions used image processing and in toolbox as shown in below figure:

			y				
	0	1	2	3	.	.	N-1
	1	*	*	*	*	*	*
	2	*	*	*	*	*	*
	3	*	*	*	*	*	*
	.	*	*	*	*	*	*
	.	*	*	*	*	*	*
x	M-1	*	*	*	*	*	*

Image processing

		c					
	1	2	3	4	.	.	N
	2	*	*	*	*	*	*
	3	*	*	*	*	*	*
	4	*	*	*	*	*	*
	.	*	*	*	*	*	*
	.	*	*	*	*	*	*
r	M	*	*	*	*	*	*

toolbox

Images as a matrices (Representing Digital Images):

The representation of an M×N numerical array in the coordinate system as:

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad A=\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

The right side of this equation is a digital image by definition. Each element of this array is called image element, or pixel. The terms image and pixel are used.
A digital image can be represented naturally as a MATLAB matrix:

$$f(x,y)=\begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \cdots & \cdots & \cdots & \cdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

Where f(1,1)=f(0,0),
Ex. F(6,2) is the element in sixth row and second column of the matrix.

- Matrix A (1,N) is called a row vector.
- (M,1) is called a column vector.
- (1,1) is called a scalar vector.

Electromagnetic Spectrum:

The electromagnetic spectrum is the range of frequencies (the spectrum) of electromagnetic radiation and their respective wave lengths and photon energies.
The electromagnetic spectrum covers electromagnetic waves with frequencies ranging from below one hertz to above 10^25 hertz, corresponding to wavelengths from thousands of kilometers down t o a fraction of the size of an atomic nucleus. This frequency range is divided into separate bands, and the electromagnetic waves within each frequency band are called by different names; beginning at the low frequency (long wavelength) end of the spectrum these are: radio waves, microwaves, terahertz waves, infrared, visible light, ultraviolet, X-rays, and gamma rays at the high-frequency (short wavelength) end as shown in figure (2). Thus digital image processing encompasses a wide and varied field of applications. Additionally, Electromagnetic (EM) energy spectrum there are sources to digital images as:
Acoustic

Ultrasonic
Electronic
Synthetic images produced by computer

.

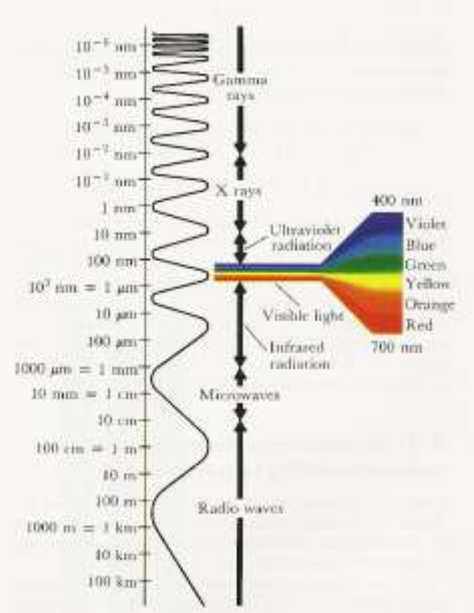
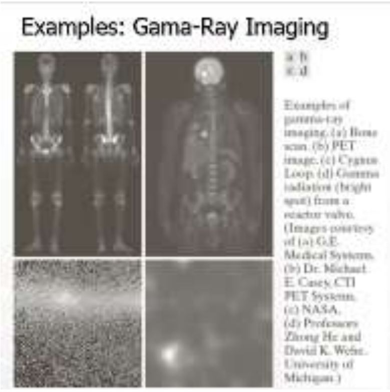


Figure (2) : Electromagnetic Spectrum (EM)

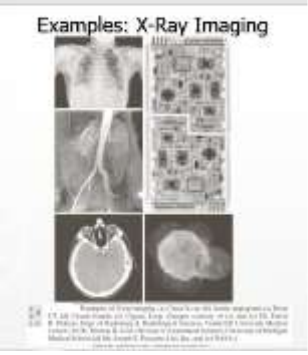
Major uses

Gamma-ray imaging: nuclear medicine and astronomical observations.
X-rays: medical diagnostics, industry, and astronomy, etc.
Ultraviolet: lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations.
Visible and infrared bands: light microscopy, astronomy, remote sensing, industry, and law enforcement.
Microwave band: radar.
Radio band: medicine (such as MRI) and astronomy.



Examples: Gama-Ray Imaging

Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation (bright spot) from a reactor valve. (Images courtesy of (a) G.E. Medical Systems, (b) Dr. Michael E. Casey, CTI PET Systems, (c) NASA, (d) Professors Zhong He and David K. Wehe, University of Michigan.)

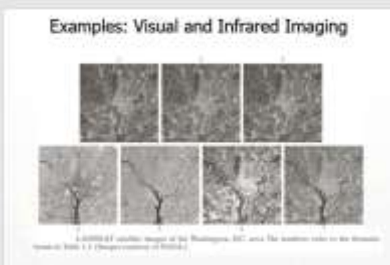


Examples: X-Ray Imaging

[illegible]

Examples: Ultraviolet Imaging

Examples of ultraviolet imaging. (a) Normal corn. (b) Smut corn. (c) Cygnus Loop. (Images courtesy of (a) and (b) Dr. Michael W. Davidson, Florida State University, (c) NASA.)



Examples: Visual and Infrared Imaging

[illegible]

Computer Imaging

Computer imaging is a fascinating and exciting area to be involved in today. The field of computer imaging separate into two primary categories:

Computer Vision and Image processing.

These two categories are not totally separate and distinct. Computer imagining can be separated into overlapping areas. Figure 1 below show that:

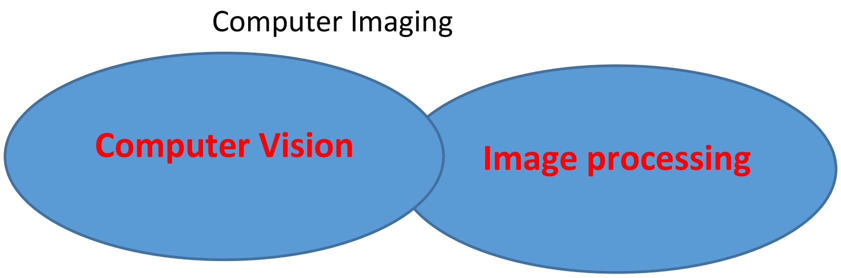


Figure (1): The overlapping between image processing and

1-Computer Vision:

Computer Vision is computer imaging, where the application does not involve a human being in the visual loop. In other word, the images are examined and acted upon by a computer. Computer vision systems used in many different areas within the medical community with only certainty being developed that the types of application will continue to grow.

2-Image Processing:

Image processing is computer imaging where the application involves a human being in the visual loop in other words the image is to be examined and acted upon by people the major to topics within the field of image processing include image restoration, image enhancement, and image compression.

There are no clear –cut bounders in the continuum from image processing at one end to computer vision at the other. Consider three types of computerized processes in these continuum:

1-Low-level processes: primitive operations ex: reduce noise, enhance contrast enhancement and sharpening.

2-Mid--level processes: segmentation (partitioning image as regions and objects).

3-High –level processes: recognizing objects.

Purpose of Image processing

The purpose of image processing is divided into 5 groups. They are:

Visualization - Observe the objects that are not visible.

Image sharpening and restoration - To create a better image.

Image retrieval - Seek for the image of interest.

Measurement of pattern – Measures various objects in an image.

Image Recognition – Distinguish the objects in an image.

Fundamental steps in Digital Image Processing:

It is convenient to subdivide different image processing algorithms into broad subclasses. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand. In other words, the steps to digital image processing as following and show that in figure 2.

1. Image Acquisition

This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling etc.

2. Image Enhancement

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as, changing brightness & contrast etc.

3. Image Restoration

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

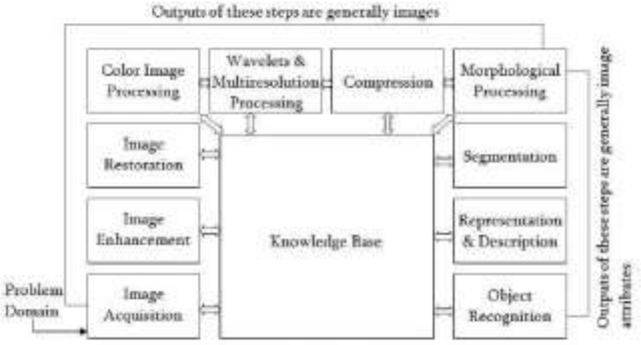


Figure (2). The fundamental steps to digital image processing.

4. Color Image Processing

Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modeling and processing in a digital domain etc.

5. Wavelets and Multiresolution Processing

Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

6. Compression

Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to compress data.

7. Morphological Processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

8. Segmentation

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

9. Representation and Description

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

10. Object recognition

Recognition is the process that assigns a label, such as, “vehicle” to an object based on its descriptors.

11. Knowledge Base:

Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications.

Some applications of digital image processing:

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a short list just to give some indication of the range of image processing applications.

1-Medicine:

Inspection and interpretation of images obtained from X-rays, MRI or CAT scans, analysis of cell images, of chromosome karyotypes.

2- Agriculture:

Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops, inspection of fruit and vegetables—distinguishing good and fresh produce from old.

3- Industry:

Automatic inspection of items on a production line, inspection of paper samples.

4- Law enforcement:

Fingerprint analysis, sharpening or de-blurring of speed-camera images.

Image Interpolation

Image interpolation is the process of using known data to estimate values at unknown location.

Image interpolation is used for zooming, shrinking (image resizing), rotating, and geometric corrections. See figures 34.

Image Interpolation sometimes called ***resampling*** *i.e.* Is technique used to create a new image— an imaging method to increase (or decrease) the number of pixels in a digital image. In other word:

- Zooming + Shrinking are image resizing tasks and come under image resampling methods.

Zooming requires two steps:

1-The creation of new pixels' location.

2-The assignment of gray level to these new locations.

Some digital cameras use interpolation to produce a larger image than the sensor captured or to create digital zoom.

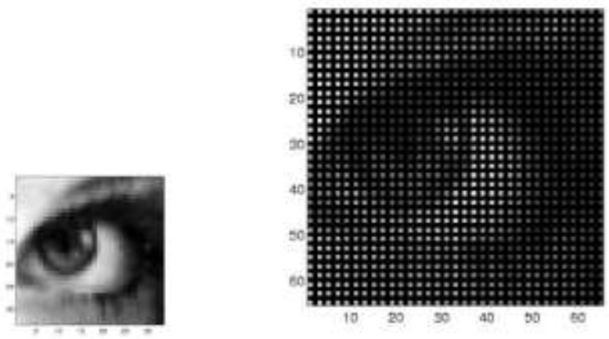


Figure 1: Image resizing (resampling)

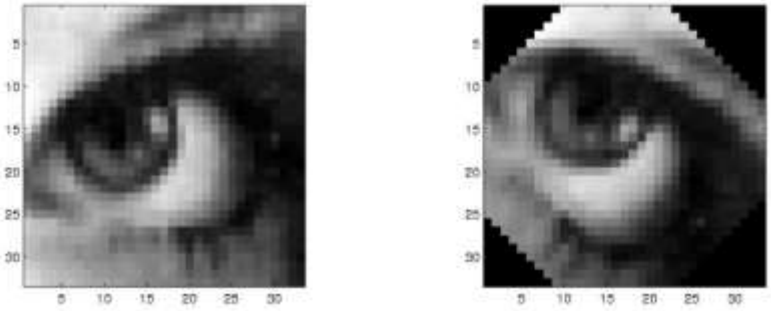


Figure 2: Image rotation

Types of interpolation:

1. Nearest neighbor interpolation(Zero-order) (See fig.3 and 4).

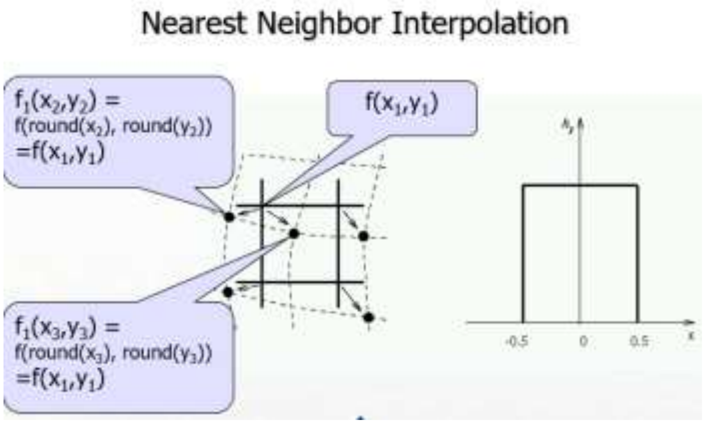

Nearest Neighbor Interpolation
$f_1(x_2,y_2) = f(\text{round}(x_2), \text{round}(y_2)) = f(x_1,y_1)$
$f(x_1,y_1)$
$f_1(x_3,y_3) = f(\text{round}(x_3), \text{round}(y_3)) = f(x_1,y_1)$
h_t
-0.5
0
0.5
x

Figure 3. Image nearest neighbor interpolation.

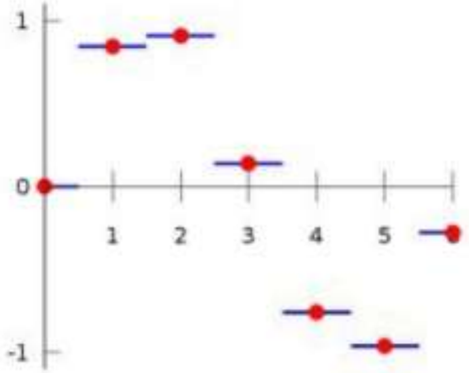


Figure 4: Relationship with 1D interpolation

W1-4pixels W2-8 pixels Complete

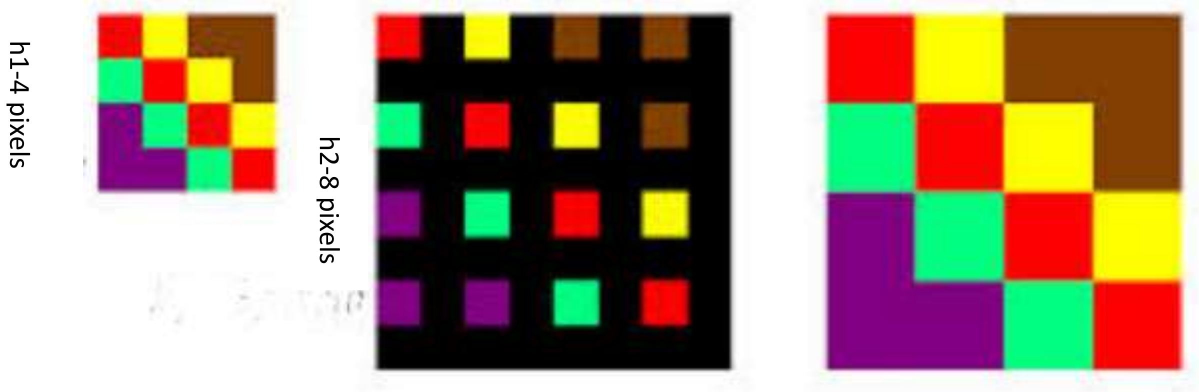


Figure 5: Image Nearest neighbor

Example: Suppose (2x2) pixels image will be enlarged be 2 times.

50	100
150	200

Original image

50	100
150	200

stretch the original to the size of the new scaled

50	50	100	100
50	50	100	100
150	150	200	200

New scaled image

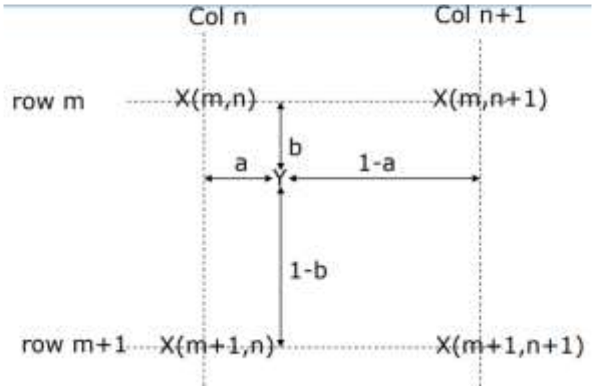
Shrinking similar to image zooming pixel replication is replaced by row & column deletion.

2-Bilinear Interpolation (First-order):

Use 4-nearset neighbors to estimate the intensity at a given location.

Let (x,y) coordinates for the pixels, and let v(x,y) denoted that value, then.(See figure 6).

Q: what is the interpolated value at Y? Ans.: (1-a)(1-b)X(m,n)+ (1-a)bX(m+1,n) +a(1-b)X(m,n+1)+abX(m+1,n+1) as shown in figure



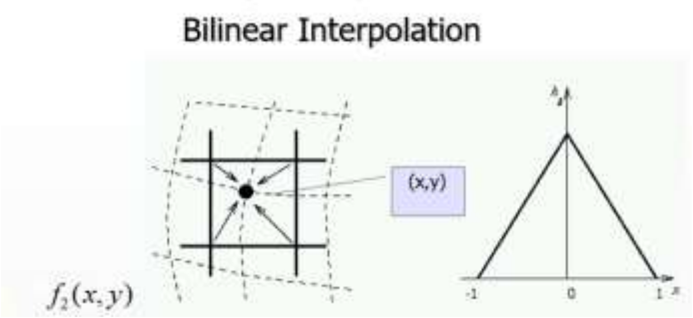


Figure 6. Image Bilinear interpolation.

In summary: Image nearest neighbor interpolation **and** Image Bilinear interpolation as shown in figure 7.

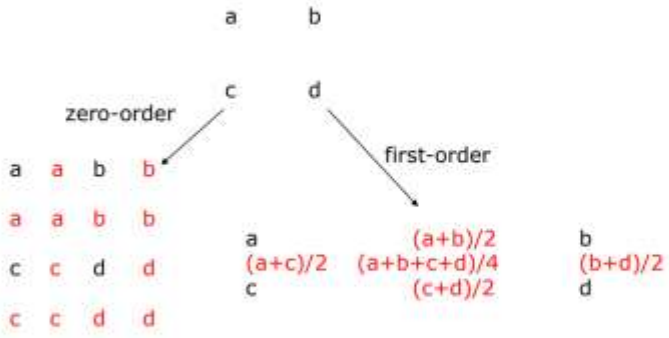


Figure 7: stretch image nearest neighbor interpolation **and** image Bilinear interpolation

3-Bicubic Interpolation (Third-order):

The sixteen coefficients are determined by using the sixteen nearest neighbors for intensity value assigned to point (x,y). As shown in figure 8.

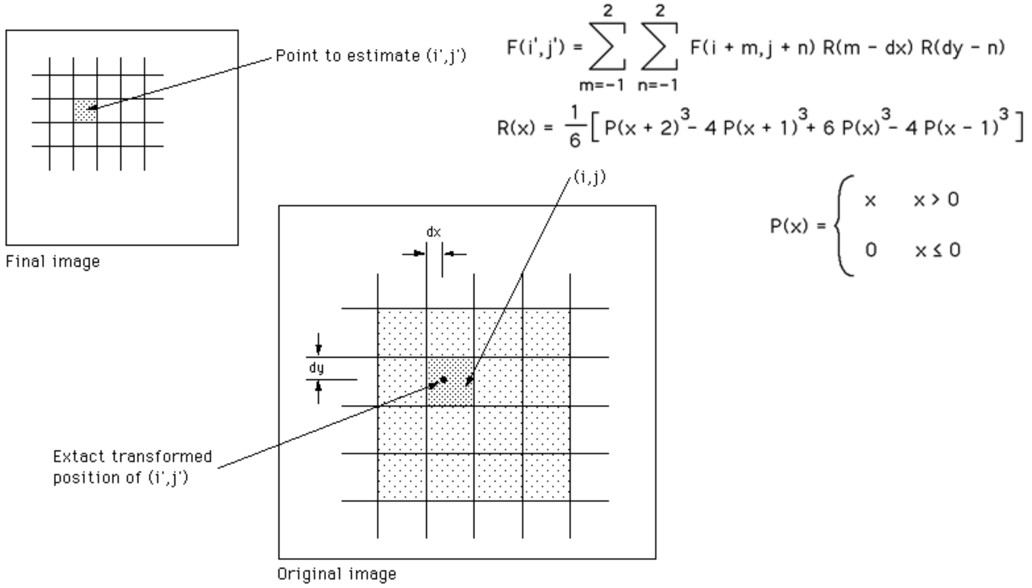

Point to estimate (i',j')
Final image
$F(i',j') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} F(i+m, j+n)\, R(m-dx)\, R(dy-n)$
$R(x) = \frac{1}{6}\left[P(x+2)^3 - 4P(x+1)^3 + 6P(x)^3 - 4P(x-1)^3 \right]$
$P(x) = \begin{cases} x & x > 0 \\ 0 & x \le 0 \end{cases}$
(i,j)
dx
dy
Extact transformed position of (i',j')
Original image

Figure 8: Bicubic Interpolation

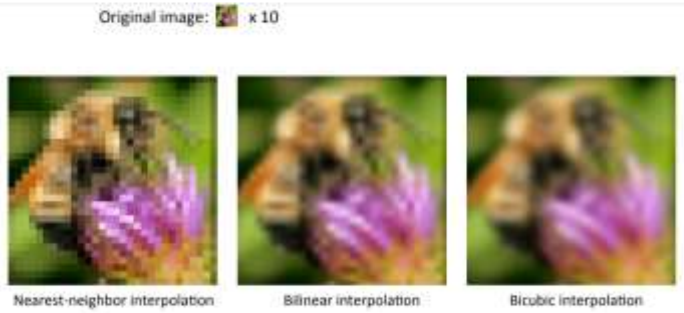


Figure 9: Examples of image interpolation.

Some Basic Relationships between Pixels :

we consider several important relationships between pixels in a digital image. When referring to a particular pixel, such as p and q.

1- Neighbors of a Pixel:

A pixel p at coordinates has four horizontal and vertical neighbors whose coordinates are given by

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

This set of pixels, called the **4-neighbors** of p, is denoted by **N_4(p)**. Each pixel is a unit distance from (x,y) , and some of the neighbor locations of p lie outside the digital image if (x,y) is on the border of the image.

The **four diagonal neighbors** of p have coordinates:

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

and are **denoted by N_D(p**). These points, together with the 4 neighbors, are called the 8-neighbors of p, denoted by N_8(p) As before, some of the neighbor locations in N_D(p) and N_8(p) **(the pixels N4(p)∪ N_D(p))** are called **8-neighbors** of p denoted **N8(p**) fall outside the image if (x,y) is on the border of the image. See figure 10.

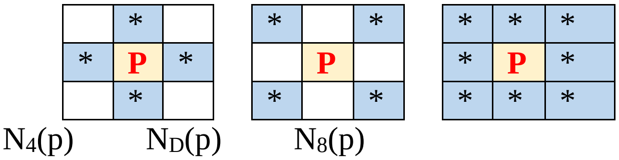


Figure 10: Set of pixels for pixel p with difference neighbors.

2-Adjacency

Let V be the set of intensity values used to define adjacency. In a binary image, if we are referring to adjacency of pixels with value 1. In a gray-scale image, the idea is the same, but set V typically contains more elements. For example, in the adjacency of pixels with a range of possible intensity values 0 to 255, set V could be any subset of these 256 values. We consider three types of adjacency:

(a) 4-adjacency: Two pixels' p and q with values from V are 4-adjacent if q is in the set.

(b) 8-adjacency: Two pixels' p and q with values from V are 8-adjacent if q is in the set.

(**c) m-adjacency (mixed adjacency**): Two pixels' p and q with values from V are m-adjacent if:

q is in N_4(p) or q is in N_D(p) and the set has no pixels whose values are from V.

3-Path

A (digital) path (or curve) from pixel p with coordinates (x_0, y_0) to pixel q with coordinates (x_n, y_n) is a sequence of distinct pixels with coordinates

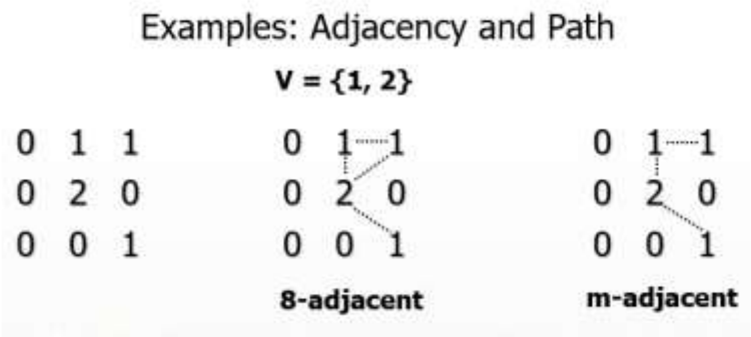
$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

Where (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \le i \le n$.

Here n is the length of the path.

If (x_0, y_0) = (x_n, y_n), the path is closed path.

We can define 4-, 8-, and m-paths based on the type of adjacency used.

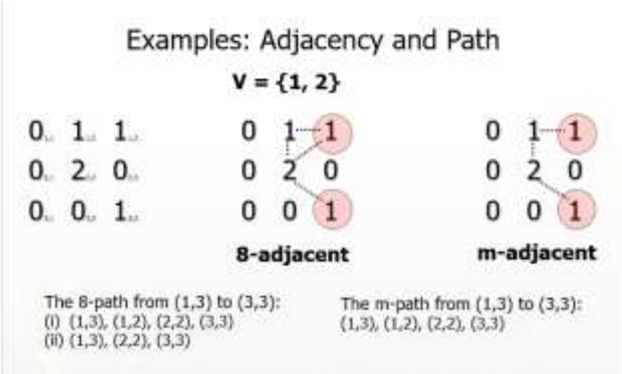


Examples: Adjacency and Path

V = {1, 2}

8-adjacent

m-adjacent



Examples: Adjacency and Path

V = {1, 2}

8-adjacent

m-adjacent

The 8-path from (1,3) to (3,3):
(i) (1,3), (1,2), (2,2), (3,3)
(ii) (1,3), (2,2), (3,3)

The m-path from (1,3) to (3,3):
(1,3), (1,2), (2,2), (3,3)

4-Connected in S

Let S represent a subset of pixels in an image. Two pixels p with coordinates (x_0, y_0) and q with coordinates (x_n, y_n) are said to be connected in S if there exists a path

(x_0, y_0), (x_1, y_1), …, (x_n, y_n)

- For every pixel *p* in S, the set of pixels in S that are connected to *p* is called a ***connected component*** of S.
- If S has only one connected component, then S is called ***Connected Set***.
- We call R a ***region*** of the image if R is a connected set.
- Two regions, R_i and R_j are said to be ***adjacent*** if their union forms a connected set.
- Regions that are not to be adjacent are said to be ***disjoint***.

Examples :

1	1	1		0	0	0	0	0		0	0	0
1	0	1	R_i	0	1	1	0	0		0	1	0
0	1	0		0	1	1	0	0		0	1	0
0	0	○		0	1	1	1	0		0	1	0
1	1	1	R_j	0	1	1	1	0		0	1	0
1	1	1		0	0	0	0	0		0	0	0

(1) Two regions (of 1s) that are adjacent if 8-adjecency is used.

(2) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used.

(3) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

Distance Measures between pixels

Given pixels p, q, and z, with coordinates (x, y), (s, t),and (v, w), respectively. The distance function D has following properties or metric if:

(a) $D(p,q) \geq 0$, $D(p,q)=0$ if $p=q$

(b) $D(p,q)= D(q,p)$

(c) $D(p,z) \leq D(q,z)+ D(p,q)$

The following are the different Distance measures:

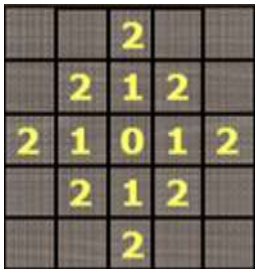
1-The Euclidean distance between p and q is defined as:

$$D_e(p,q)=\{(x-s)^2+(y-t)^2\}^{1/2}$$

For this distance measure, the pixels having a distance less than or equal to some value r from (x, y) are the points contained in a disk of radius r centered at (x, y).

2- The City -Block Distance

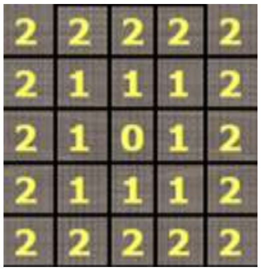
D4(p, q) = |x-s| + |y-t|



		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

3- The Chessboard Distance

D8(p, q) = max(|x-s|, |y-t|)

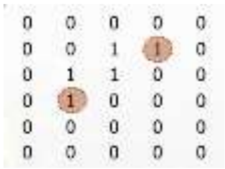


2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Q1\\In the following arrangement of pixels, what’s the value of the chessboard distance between the circled two points?

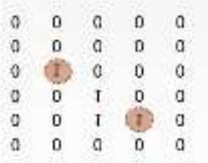
Q2\\In the following arrangement of pixels, what’s the value of the city-block distance between the circled two points?

Q3\\ In the following arrangement of pixels, what’s the value of the length of the m-path between the circled two points?



0	0	0	0	0
0	0	1	1	0
0	1	1	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Q4\\ In the following arrangement of pixels, what’s the value of the length of the m-path between the circled two points?

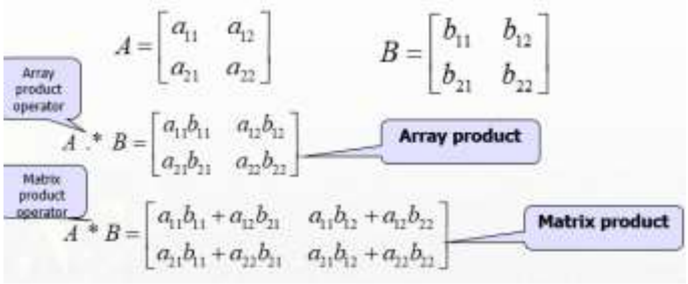


0	0	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	1	1	0
0	0	0	0	0

Mathematical Tools Used in Digital Image Processing:

1-Array versus Matrix Operations:

An array operation involving one or more images is carried out on a pixel-by pixel basis. For example, consider the following images:



$$A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$A\ .*\ B=\begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

$$A * B=\begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}$$

2-Linear versus Nonlinear Operations:

One of the most important classifications of an image-processing method is whether it is linear or nonlinear. Consider a general operator, H, that produces an output image, g(x, y),for a given input image, f(x, y):

$$H\left[f(x,y)\right]=g(x,y)$$

$$\begin{aligned} &H\left[a_i f_i(x,y)+a_j f_j(x,y)\right] \\ &=H\left[a_i f_i(x,y)\right]+H\left[a_j f_j(x,y)\right] \\ &=a_i H\left[f_i(x,y)\right]+a_j H\left[f_j(x,y)\right] \\ &=a_i g_i(x,y)+a_j g_j(x,y) \end{aligned}$$

H is said to be a **linear operator;**

H is said to be a **nonlinear operator** if it does not meet the above qualification.

3-Arithmetic operations:

Arithmetic operations between images are array operations. The four arithmetic operations are denoted as

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

$$d(x,y) = f(x,y) - g(x,y)$$

$$p(x,y) = f(x,y) \times g(x,y)$$

$$v(x,y) = f(x,y) \div g(x,y)$$

Example: Addition of noisy images for noise reduction.

Noiseless image: f(x,y)

Noise: n(x,y) (at every pair of coordinates (x,y), the noise is uncorrelated and has zero average value)

Corrupted image: g(x,y)

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

Reducing the noise by adding a set of noisy images, $g_i(x,y)$

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

$$E\left\{\bar{g}(x,y)\right\} = E\left\{\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)\right\}$$

$$= E\left\{\frac{1}{K}\sum_{i=1}^{K}\left[f(x,y) + n_i(x,y)\right]\right\}$$

$$= f(x,y) + E\left\{\frac{1}{K}\sum_{i=1}^{K} n_i(x,y)\right\}$$

$$= f(x,y)$$

-In astronomy, imaging under very low light levels frequently causes sensor noise to render single images virtually useless for analysis.

-In astronomical observations, similar sensors for noise reduction by observing the same scene over long periods of time. Image averaging is then used to reduce the noise. See figure 1.

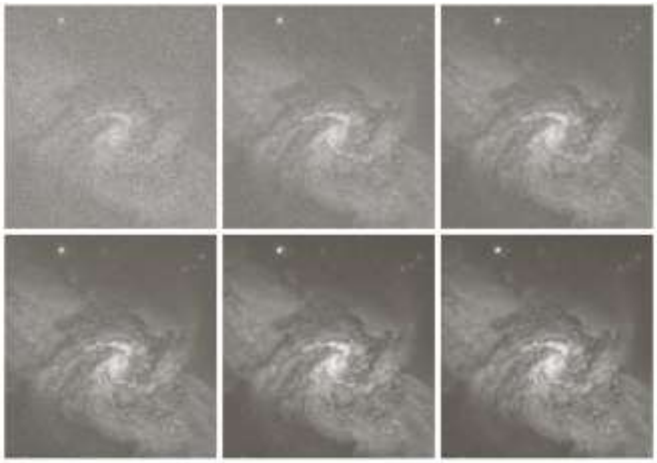


FIGURE 1 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Example of Image Subtraction: Mask Mode Radiography

Mask h(x,y): an X-ray image of a region of a patient's body

Live images f(x,y): X-ray images captured at TV rates after injection of the contrast medium

Enhanced detail g(x,y)

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

The procedure gives a movie showing how the contrast medium propagates through the various arteries in the area being observed. See figure 2.

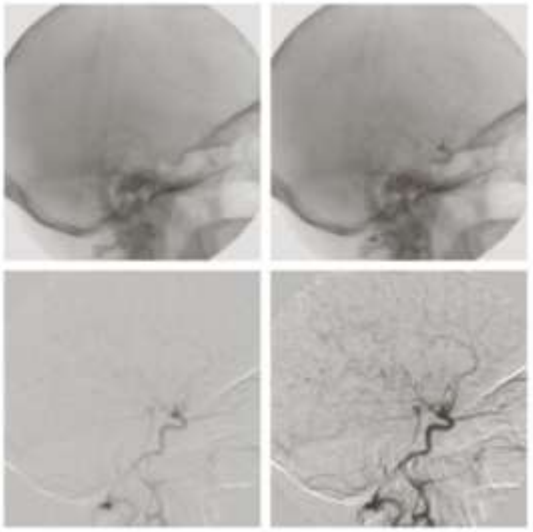


FIGURE 2 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

An Example of Image Multiplication

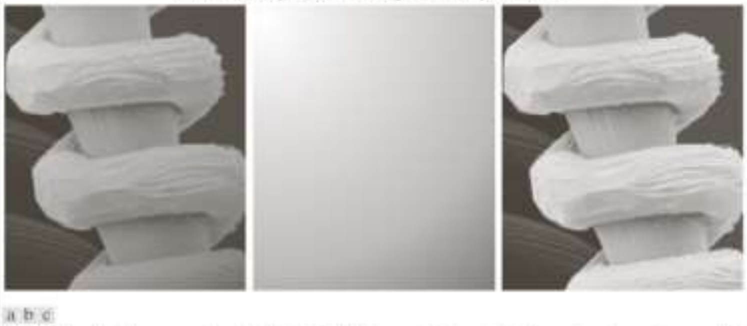


FIGURE 3 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Set and Logical Operations

1-Basic set operations:

Let A be a set composed of ordered pairs of real numbers. If is an element of A, then we write

$$a \in A$$

Similarly, if a is not an element of A, we write

$$a \notin A$$

The set with no elements is called the **null** or **empty** set and is denoted by the symbol Ø

If every element of a set A is also an element of a set B, then A is said to be a subset of B, denoted as

$$A \subseteq B$$

The union of two sets A and B, denoted by

$$C = A \cup B$$

Similarly, the intersection of two sets A and B, denoted by

$$D = A \cap B$$

Two sets A and B are said to be **disjoint** or **mutually** exclusive if they have no common elements

$$A \cap B = \varnothing$$

The set universe, U, is the set of all elements in a given application. In image processing, we typically define the universe to be the rectangle containing all the pixels in an image.

The **complement of a set** A is the set of elements that are not in A:

$$A^c = \{w \mid w \notin A\}$$

The difference of two sets A and B, denoted is defined as

$$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$$

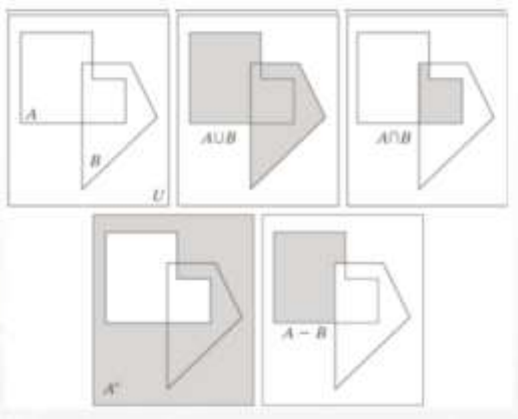
define Ac in terms of U and the set difference operation:

$$A^c = U - A.$$

the union and intersection operations for gray-scale values usually are defined as the max and min of corresponding pixel pairs, respectively, while the complement is defined as the pairwise differences between a constant and the intensity of every pixel in an image.

Figure 1 illustrates the preceding concepts of the set operation indicated in each figure is shown in gray:

Set and Logical Operations



a b c
d e

FIGURE 1
(a) Two sets of coordinates, *A* and *B*, in 2-D space. (b) The union of *A* and *B*. (c) The intersection of *A* and *B*. (d) The complement of *A*. (e) The difference between *A* and *B*. In (b)–(e) the shaded areas represent the member of the set operation indicated.

Let A be the elements of a gray-scale image. The elements of A are triplets of the form (x, y, z), where x and y are spatial coordinates and z denotes the intensity at the point (x, y).

$$A \;= \{(x, y, z) \,|\, z = f(x, y)\}$$

The complement of A is denoted Ac

$$A^c = \{(x, y, K - z) \,|\, (x, y, z) \in A\}$$

$$K = 2^k - 1; k \text{ is the number of intensity bits used to represent } z$$

Example 1: Find **A**^{c} for sub gray level with 8 bits.

A=

15	167	200	0
1	35	133	14
188	12	204	44
78	24	10	255

A^c =(x,y,255-z|**A**)

A^c **=**

240	88	55	255
254	220	122	241
67	243	51	211
177	231	245	0

The union of two gray-scale images (sets) A and B is defined as the set

$$\boldsymbol{A} \cup \boldsymbol{B} = \{\max(a, b) | a \in \boldsymbol{A}, b \in \boldsymbol{B}\}$$

Example 2:

A=

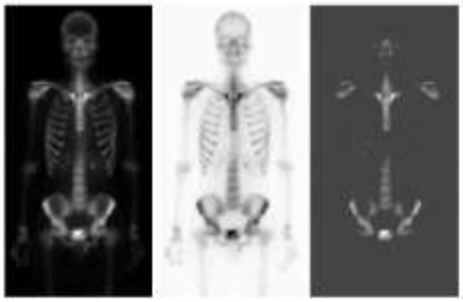
15	167	200	0
1	35	133	14
188	12	204	44
78	24	10	28

B=

10	230	0	9
12	24	133	111
88	0	245	144
34	234	10	228

$\boldsymbol{A} \cup \boldsymbol{B} =$

15	230	200	9
12	35	133	111
188	12	245	144
78	234	10	228



a b c

FIGURE 2. Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

2-Logical operations:

When dealing with binary images, we can think of foreground (1-valued) and background (0-valued) sets of pixels.

OR, AND, and **NOT** logical operations:

Consider two regions (sets) A and B composed of foreground pixels.

The **OR** of these two sets is the set of elements (coordinates) belonging either to A or B or to both.

The **AND** operation is the set of elements that are common to A and B.

The **NOT** operation of a set A is the set of elements not in A. Because we are dealing with images, if A is a given set of foreground pixels, NOT(A) is the set of all pixels in the image that are not in A.

XOR (exclusive **OR**) operation, which is the set of foreground pixels belonging to A or B, but not both.

Example 3:

1)Find **AND**, **OR** for two binary images **A** and **B** .

2)Find $\mathbf{A}^c$.

A **B** ***Result***

1	0	1	**AND**	0	1	1	=	0	0	1
1	1	0		0	1	0		0	1	0

1	0	1	**OR**	0	1	1	=	1	1	1
1	1	0		0	1	0		1	1	0

A **A^c**

1	0	1	**NOT**	0	1	0
1	1	0		0	0	1

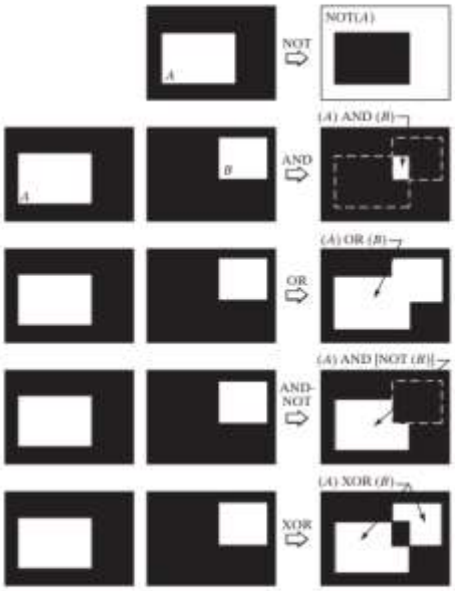


FIGURE 3 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Problems:

1-Find **AND** , **OR**, and **NOT** for two images:

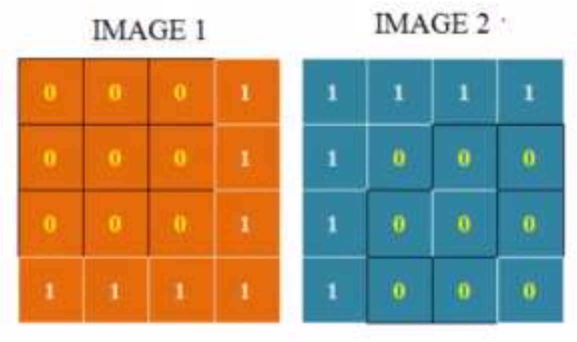


IMAGE 1

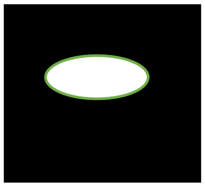
0	0	0	1
0	0	0	1
0	0	0	1
1	1	1	1

IMAGE 2

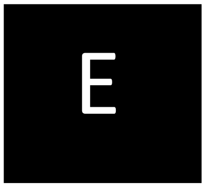
1	1	1	1
1	0	0	0
1	0	0	0
1	0	0	0

2-Find A^c **,** A **OR** B**,** A **AND** B**, and** A **XOR** B for two binary images A and B.

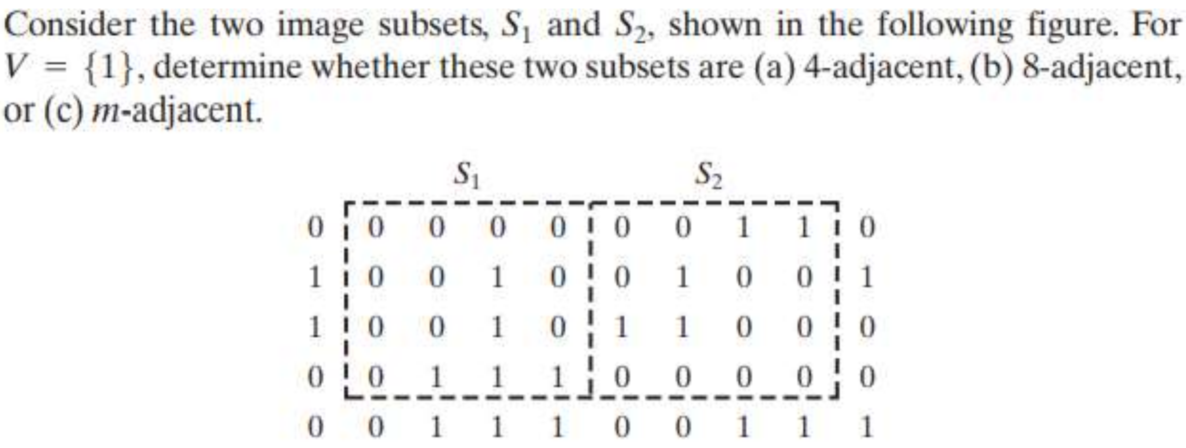
Binary Image-Image A



Binary Image-Image B



3-



Consider the two image subsets, S_1 and S_2, shown in the following figure. For $V = \{1\}$, determine whether these two subsets are (a) 4-adjacent, (b) 8-adjacent, or (c) m-adjacent.

	S_1				S_2				
0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1

4- Consider the image segment shown.

★**(a)** Let $V = \{0, 1\}$ and compute the lengths of the shortest 4-, 8-, and m-path between p and q. If a particular path does not exist between these two points, explain why.

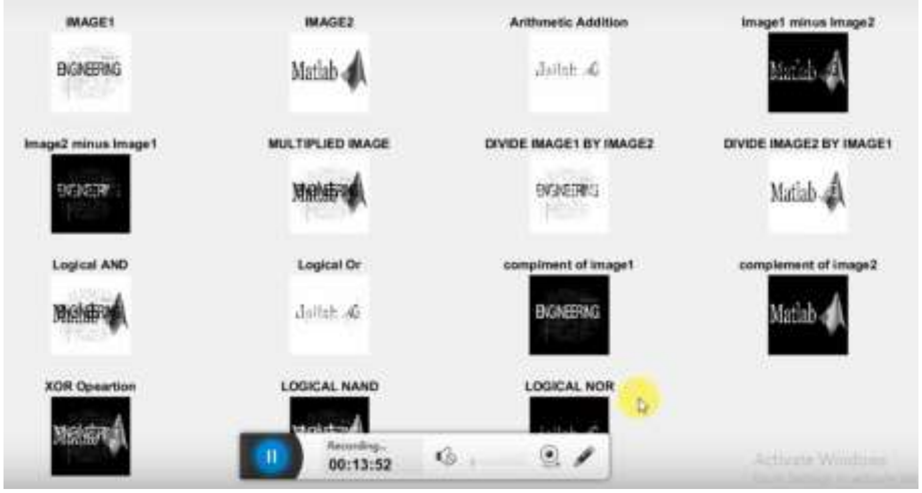
(b) Repeat for $V = \{1, 2\}$.

$$\begin{array}{rcccl} & 3 & 1 & 2 & 1\,(q) \\ & 2 & 2 & 0 & 2 \\ & 1 & 2 & 1 & 1 \\ (p) & 1 & 0 & 1 & 2 \end{array}$$

(**c**) Given the condition (s) under which the D_4 distance between two points p and q is equal to the shortest 4-path between these points.

(**d**) Is this path unique?

Repeat problem 3 for the D_8 distance.



Spatial Operations

Spatial operations classify into three broad categories:

1- single-pixel operations.

2- neighborhood operations

3-geometric spatial transformations.

1-Single-pixel operations:

This type of process may be expressed as a transformation function, T, of the form:

$$s = T(z)$$

where z is the intensity of a pixel in the original image and s is the (mapped) intensity of the corresponding pixel in the processed image.

Ex.: Operator as a negative image with 8 bits. See figure 1.

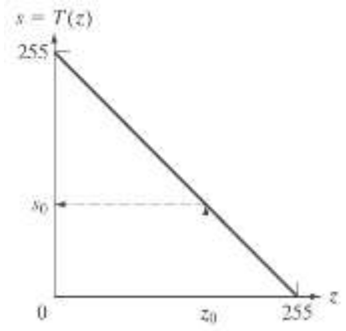


Figure 1: Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0 .

2-Neighborhood operations:

Let S_{xy} denote the set of coordinates of a neighborhood centered on an arbitrary point (x, y) in an image, f. The value of this pixel is determined by a specified operation involving the pixels in the input image with coordinates in S_{xy}

Ex.: Average value of the pixels in a rectangular neighborhood of size m x n centered on (x, y).

We can express this operation in equation form as:

$$g(x, y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} f(r, c)$$

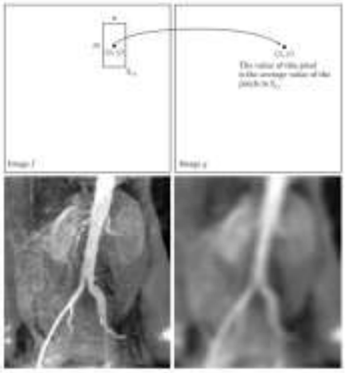
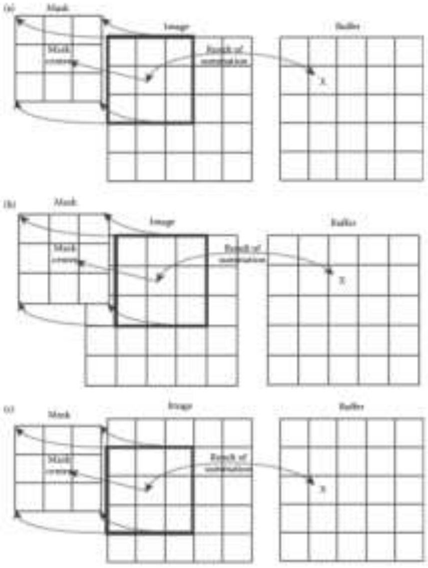

n
m
(x, y)
S_{xy}
(x, y)
The value of this pixel is the average value of the pixels in S_{xy}
Image f
Image g

Figure 2. Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram (d) The result of The images (790 x 686) pixels with m = n = 41 by above equation g(x,y).



3- Geometric spatial transformations:

Geometric transformations (rubber-sheet transformation) modify the spatial relationship between pixels in an image. a geometric transformation consists of two basic operations:

1- Spatial transformation of coordinates.

2- Intensity interpolation.

intensity interpolation that assigns intensity values to the spatially transformed pixels.

The transformation of coordinates may be expressed as:

$$(x, y) = T\{(v, w)\}$$

where (v,w) are pixel coordinates in the original image and (x, y) are the corresponding pixel coordinates in the transformed image. . For example, the transformation (v/2, w/2) shrinks the original image to half its size in both spatial directions

One of the most commonly used spatial coordinate transformations is the affine transform

which has the general form:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

This transformation can scale, rotate, translate, or sheer a set of coordinate points, depending on the value chosen for the elements of matrix **T,** see table 1. The preceding transformations relocate pixels on an image to new locations.

we can use above Equation (affine transform) in two basic ways. The first, called a ***forward mapping*** as:

$$(x, y) = T\{(v, w)\}$$

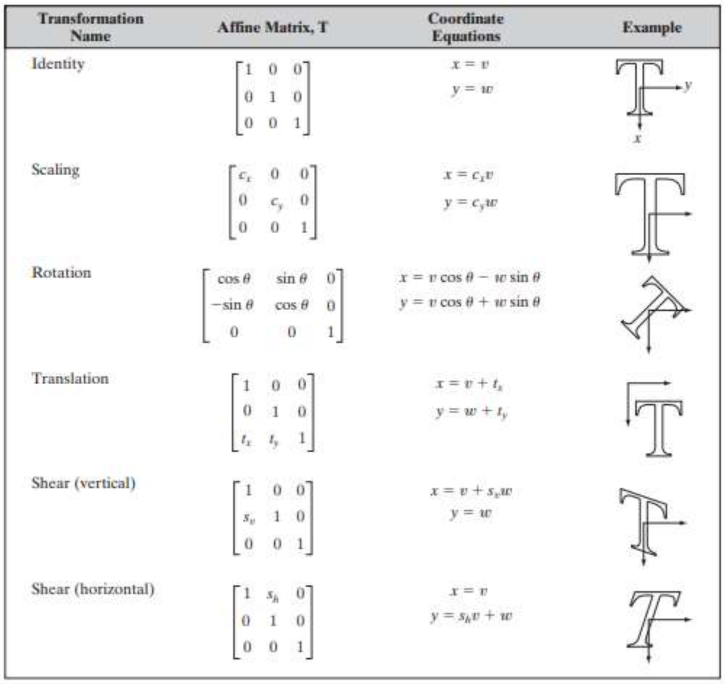
It’s possible that two or more pixels can be transformed to the same location in the output image.

The second approach, called ***inverse mapping***,

$$(v, w) = T^{-1}\{(x, y)\}$$

The nearest input pixels to determine the intensity of the output pixel value. Inverse mappings are more efficient to implement than forward mappings.

Table 1: Affine transform.



Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix}c_x & 0 & 0\\0 & c_y & 0\\0 & 0 & 1\end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix}\cos\theta & \sin\theta & 0\\-\sin\theta & \cos\theta & 0\\0 & 0 & 1\end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\t_x & t_y & 1\end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix}1 & 0 & 0\\s_v & 1 & 0\\0 & 0 & 1\end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix}1 & s_h & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$	$x = v$ $y = s_h v + w$	

Image registration:

Image registration is the process of transforming different sets of data into one coordinate system. Data may be multiple photographs, data from different sensors, times, depths, or viewpoints.

One of the principal approaches for image registration is to use ***tie points*** (also called ***control points***)

The corresponding points are known precisely in the input and output (**reference**) images.

Goal: Estimate the transformation function and use it to register the two images.

The use of image registration:

Image registration could be used in two ways:

1-Image-Image Registration: Two or more pictures are aligned to integrate or fuse matching pixels representing the same objects.

2-Image to Map Registration: In this step, the input image is distorted to match the map information of a base image while keeping its original spatial resolution

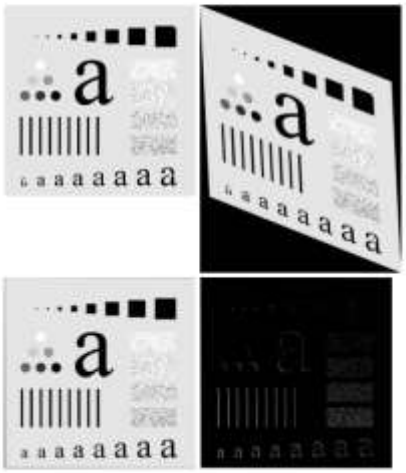
Applications:

- It is an important application of digital image processing.

-It is used in computer vision, medical imaging, military automatic target recognition, and compiling and analyzing images and data from satellites. Registration is necessary in order to be able to compare or integrate the data obtained from these different measurements.

The input and output images are available but the transformation function is unknown.

In medical images: registration (for data of the same patient taken at different points in time such as change detection or tumor monitoring). .



a b
c d

FIGURE .3 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the border). (d) Difference between (a) and (c), showing more registration errors.

Image Enhancement

Image enhancement belongs to image preprocessing methods.

Objective of image enhancement – process:

*The principal objective of image enhancement is to process a given image so that the result is more suitable than the original image for a specific application.

*It sharpens image features such as edges, boundaries, or contrast to make a graphic display more helpful for display and analysis.

*The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily.

*The greatest difficulty in image enhancement is quantifying the criterion for enhancement.

* Image enhancement methods can be based on either spatial or frequency domain techniques.

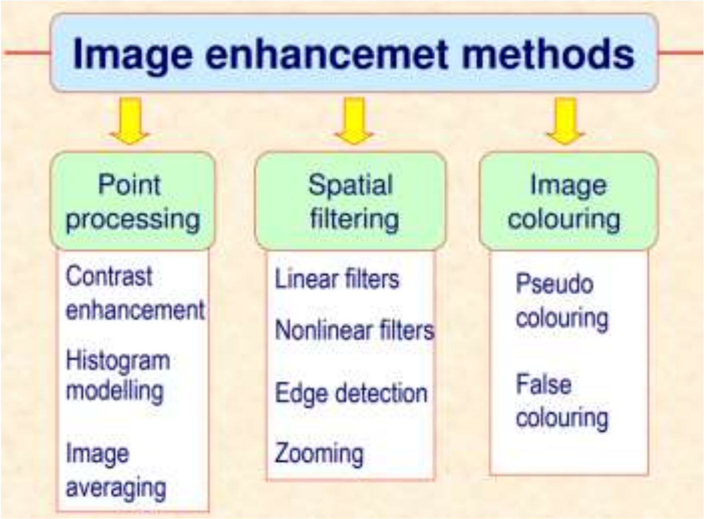


Image enhancemet methods

Point processing
- Contrast enhancement
- Histogram modelling
- Image averaging

Spatial filtering
- Linear filters
- Nonlinear filters
- Edge detection
- Zooming

Image colouring
- Pseudo colouring
- False colouring

Simple intensity transformation:

1-Image negatives:

Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing.

Transform function T : g(x,y)=L-f(x,y), where L is the max. intensity.

2-Contrast stretching:

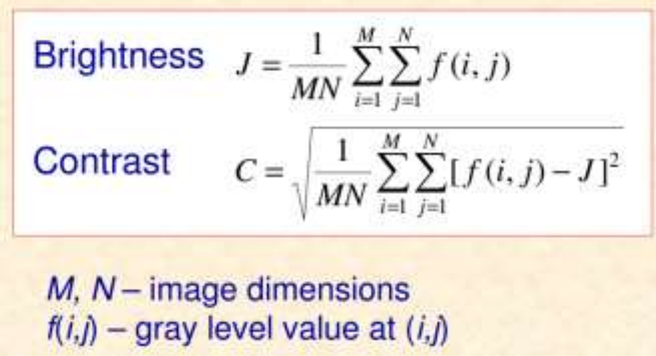
Low-contrast images can result from poor illumination, lack of dynamic range in the image sensor. The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.

3-Compression of dynamic range:

An effective way to compress the dynamic range of pixel values is to perform the following intensity transformation function: s = c log(1+|r|) where c is a scaling constant, and the logarithm function performs the desired compression.

4-Gray-level slicing:

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in x-ray images.



Brightness $J = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)$

Contrast $C = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j) - J]^2}$

M, N – image dimensions
f(*i,j*) – gray level value at (*i,j*)

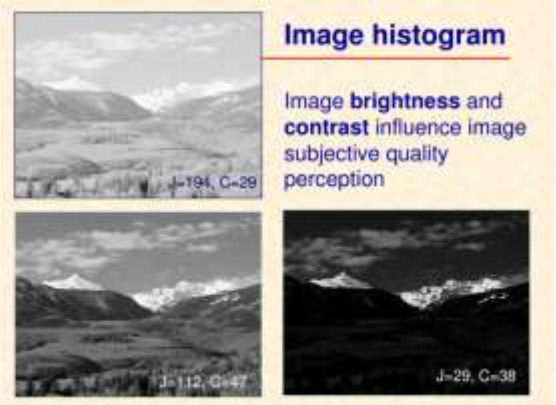


Image histogram

Image **brightness** and **contrast** influence image subjective quality perception

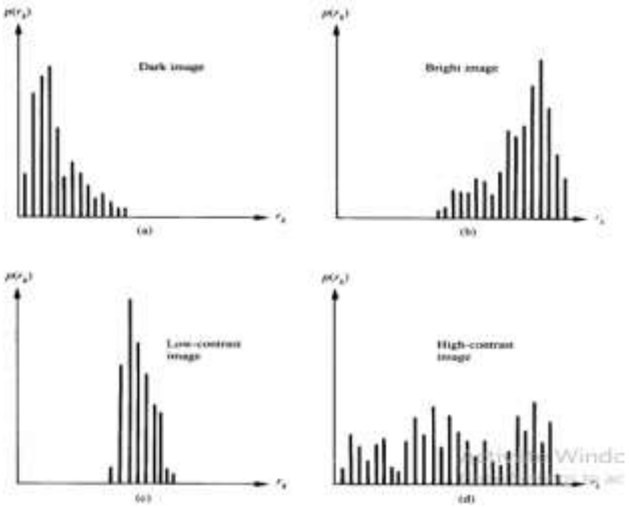
Histogram processing:

*The histogram of a digital image with gray levels in the range [0,L-1] is a discrete function $p(r_k)=n_k/n$, where r_k is the kth gray level, n_k is the number of pixels in the image with that gray level, n is the total number of pixels in the image, and k=0,1..L-1.

* P(r_k) gives an estimate of the probability of occurrence of gray level r_k.

*The shape of the histogram of an image gives us useful information about the possibility for contrast enhancement.

* A histogram of a narrow shape indicates little dynamic range and thus corresponds to an image having low contrast.



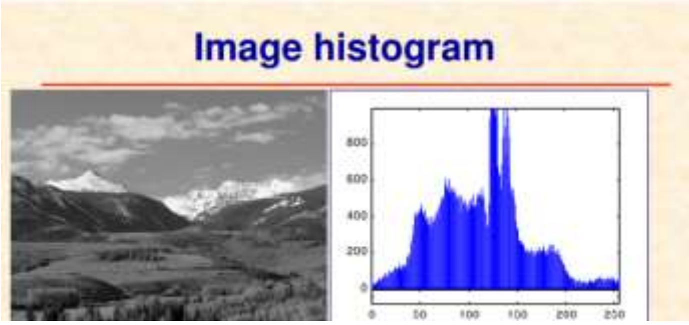


Image histogram

Spatial Filtering

Spatial filtering may be considered as an extension of point processing, where we apply a function to a neighborhood of each pixel. The idea is to move a “mask”: a rectangle or other shape over the given image. we create a new image whose pixels have grey values calculated from the grey values under the mask, as shown in figure (1). The combination of mask and function is called filter

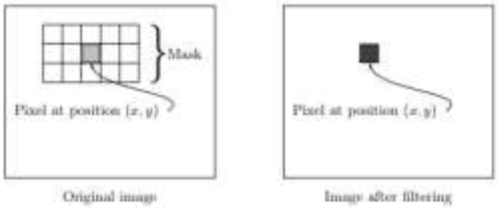
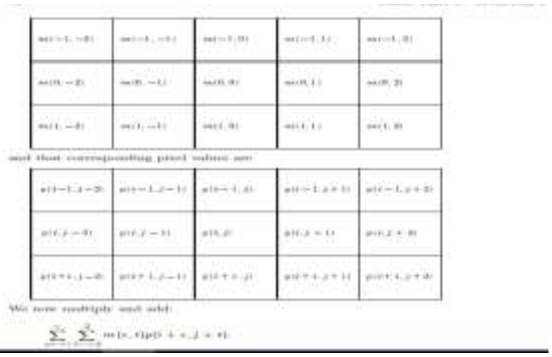


Figure (1): Using a spatial mask on an image.

If the function by which the new grey value is calculated is a linear function of all the grey values in the mask, then the filter is called a **linear filter**. Suppose that the mask values are given by:



$m(-1,-2)$	$m(-1,-1)$	$m(-1,0)$	$m(-1,1)$	$m(-1,2)$
$m(0,-2)$	$m(0,-1)$	$m(0,0)$	$m(0,1)$	$m(0,2)$
$m(1,-2)$	$m(1,-1)$	$m(1,0)$	$m(1,1)$	$m(1,2)$

and that corresponding pixel values are:

$p(i-1,j-2)$	$p(i-1,j-1)$	$p(i-1,j)$	$p(i-1,j+1)$	$p(i-1,j+2)$
$p(i,j-2)$	$p(i,j-1)$	$p(i,j)$	$p(i,j+1)$	$p(i,j+2)$
$p(i+1,j-2)$	$p(i+1,j-1)$	$p(i+1,j)$	$p(i+1,j+1)$	$p(i+1,j+2)$

We now multiply and add:

$$\sum_{s=-1}^{1}\sum_{t=-2}^{2} m(s,t)p(i+s,j+t)$$

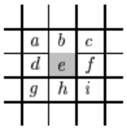
Figure (2): Diagram of spatial filtering.

A diagram illustrating the process for performing spatial filtering is given in figure 2. We see that spatial filtering requires three steps:

1. position the mask over the current pixel.
2. form all products of filter elements with the corresponding elements of the neighborhood.
3. add up all the products.

This must be repeated for every pixel in the image.

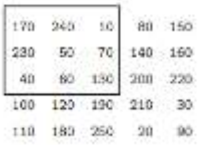
We may describe this operation as follow:



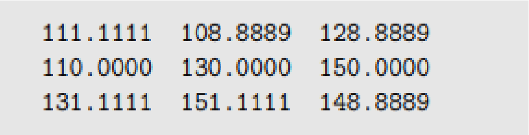
$$\longrightarrow \frac{1}{9}(a+b+c+d+e+f+g+h+i)$$

Where the letters a,b,c…..,I are the values of gray levels for the pixels in the mask. The average is the grey value of the corresponding pixel in the new image.

Example 1: To apply this to an image, consider the 5x5 "image" with 3x3 mask.



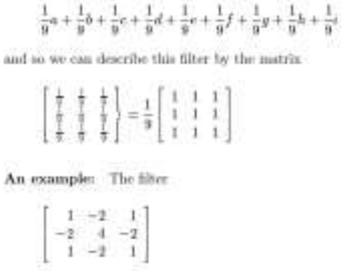
170	240	10	80	150
230	50	70	140	160
40	80	130	200	220
100	120	190	210	30
110	180	250	20	90



```
111.1111   108.8889   128.8889
110.0000   130.0000   150.0000
131.1111   151.1111   148.8889
```

The result is:

It is convenient to describe a linear filter simply in terms of the coefficients of all the grey values of pixels within the mask:



$$\frac{1}{9}a+\frac{1}{9}b+\frac{1}{9}c+\frac{1}{9}d+\frac{1}{9}e+\frac{1}{9}f+\frac{1}{9}g+\frac{1}{9}h+\frac{1}{9}i$$

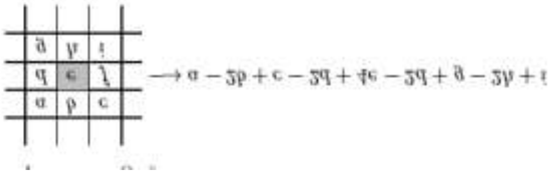
and so we can describe this filter by the matrix

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

An example: The filter

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Would operate on grey values as



Edge of the image:

There are a number of different approaches to dealing with edge of image.

1- **Ignore the edges**:
 we only apply all pixels except for the edges, and results in an output image which is smaller than the original. In this method lose a significant amount of information.

2- **"Pad" with zeros:**
 We assume that all necessary values outside the image are zero.
Figure (3) explain the edge of image.

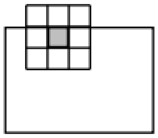
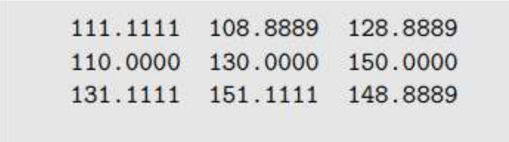


Figure (3): A mask at the edge of an image.

1-Applies the mask only to “**inside” pixels**. The image in example 1. We can see that the result will always be smaller than the original. It used ‘**’ignore the edge** ‘’by using the filter:

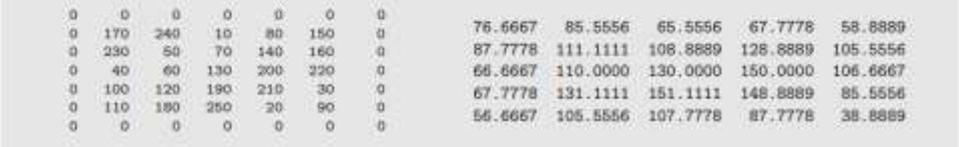
$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

We get:



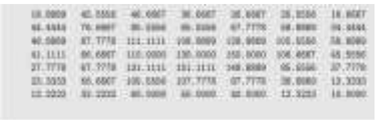
```
111.1111   108.8889   128.8889
110.0000   130.0000   150.0000
131.1111   151.1111   148.8889
```

2- It produces a matrix of **equal size (same)** to the original image matrix. It uses **zero padding:**



```
  0     0     0     0     0     0     0        76.6667    85.5556    65.5556    67.7778    58.8889
  0   170   240    10    80   150     0        87.7778   111.1111   108.8889   128.8889   105.5556
  0   230    50    70   140   160     0        66.6667   110.0000   130.0000   150.0000   106.6667
  0    40    60   130   200   220     0        67.7778   131.1111   151.1111   148.8889    85.5556
  0   100   120   190   210    30     0        56.6667   105.5556   107.7778    87.7778    38.8889
  0   110   180   250    20    90     0
  0     0     0     0     0     0     0
```

3-it used larger **than the origin(full)** by **padding with zero**, and applying the filter at all places on and around the image where the mask intersects the image matrix.



```
 18.8889    45.5556    46.6667    36.6667    26.6667    25.5556    16.6667
 44.4444    76.6667    85.5556    65.5556    67.7778    58.8889    34.4444
 48.8889    87.7778   111.1111   108.8889   128.8889   105.5556    58.8889
 41.1111    66.6667   110.0000   130.0000   150.0000   106.6667    45.5556
 27.7778    67.7778   131.1111   151.1111   148.8889    85.5556    37.7778
 23.3333    56.6667   105.5556   107.7778    87.7778    38.8889    13.3333
 12.2222    32.2222    60.0000    50.0000    40.0000    12.2222    10.0000
```

Frequencies; low and high pass filters:

One important aspect of an image which enables us to do this is the notion of frequencies. Fundamentally, the **frequencies** of an image are the amount by which grey values change with distance. **High frequency components** are characterized by large changes in grey values over small distances; example of high frequency components are edges and noise. **Low frequency components**, on the other hand, are parts of the image characterized by little change in the grey values. These may include backgrounds, skin textures.

We then say that a filter is a

High pass filter if it “passes over” the high frequency components, and reduces or eliminates low frequency components.

Low pass filter if it “passes over” the low frequency components, and reduces or eliminates high frequency component.

The 3x3 averaging filter is *low pass filter*, as it tends to blur edges.

The filter is a high pass filter as:

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

We note that the sum of the coefficients (that is, the sum of all e elements in the matrix), in the high pass filter is zero. This means that in a low frequency part of an image, where the grey values are similar.

Example 2: consider a 4x4 block of similar values pixels, and apply the above high pass filter to the center four.

150	152	148	149
147	152	151	150
152	148	149	151
151	149	150	148

$\longrightarrow$

11	6
−13	−5

The resulting values are close to zero.

High pass filters are of particular value in edge detection and edge enhancement.

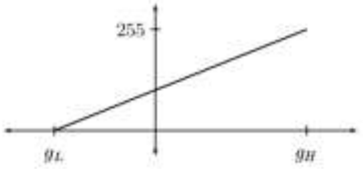
Values outside the range 0–255 :

$$y = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \le x \le 255 \\ 255 & \text{if } x > 255 \end{cases}$$

This will produce an image with all pixel values in the required range, but is not suitable if there are many grey values outside the 0–255 range.

Scaling transformation:

Suppose the lowest grey value produced by the filter if g_L and the highest value is g_H. We can transform all values in the range (g_L-g_H) the range 0–255 by the linear transformation illustrated below.



Since the gradient of the line is 255/((g_H-g_L) we can write the equation:

$$y = 255\frac{x - g_L}{g_H - g_L}$$

and applying this transformation to all grey levels x produced by the filter will result (after any necessary rounding) in an image which can be displayed.

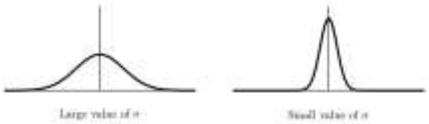
Gaussian filters:

Gaussian filters are a class of low-pass filters, all based on the Gaussian probability distribution

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}$$

function

Where σ is the standard deviation: a large value of σ produces to a flatter curve, and a small value leads to a “pointier” curve filter.



A two dimensional Gaussian function is given

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

Gaussian filters have a blurring effect which looks very similar to that produced by neighborhood averaging. Although the results of Gaussian blurring and averaging look similar, the Gaussian filter has some elegant mathematical properties which make it particularly suitable for blurring. We see that to obtain a spread out blurring effect, we need a large standard deviation. In fact, if we let the standard deviation grow large without bound, we obtain the averaging filters as limiting values.

References:

1- Gonzales R.C. and Woods P., Digital Image Processing, Addison- Wesley,4th edition, 2018

2-Alasdair McAndrew, An Introduction to Digital Image Processing with Matlab Notes for SCM2511 Image Processing