University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Image Processing 2
معالجة صور 2

Lect. Prof. Dr. Nidaa Flaih Hassan
أ. د. نداء فليح حسن

cs.uotechnology.edu.iq

Course 2 Outline: Image Processing 2

Chapter One: Histogram of Image

- Histogram Modification
  - Stretching mapping function
  - Shrinking mapping function
  - Sliding Technique.
- Histogram Equalization
- Histogram Features.

Chapter Two: Image Segmentation

- Region-based segmentation
- Clustering method.
- Boundary detection and Combined approaches.

Chapter Three: Image Compression

- Compression System Model
  - Compression Ratio and Entropy
- Lossless Compression Method
  - Run Length Coding
  - Huffman Coding

Chapter Four: Discrete transform

- Fourier Transform.
- Cosine Transform.
- Wavelet Transform

Chapter Five: Image Fidelity Criteria

- Objective fidelity criteria
- Subjective fidelity criteria

Chapter One
Histogram of Image

1.1 Introduction to Histogram

The histogram of an image is a plot of the gray _levels values versus the number of pixels at that value.

A histogram appears as a graph with "brightness" on the horizontal axis from 0 to 255 (for an 8-bit) intensity scale) and "number of pixels "on the vertical axis. For each colored image three histograms are computed, one for each component (RGB, HSL). The histogram gives us a convenient -easy -to -read representation of the concentration of pixels versus brightness of an image, using this graph we able to see immediately:

1 Whether an image is a dark or light, and high or low contrast.
2 Give us our first clues about what contrast enhancement would be appropriately applied to make the image more subjectively pleasing to an observer, or easier to interpret by succeeding image analysis operations.

So the shape of the histogram provides us with information about the nature of the image or sub-image if we considering an object within the image. For example:

1 A very narrow histogram implies a low-contrast image
2 Histogram skewed (مائل) to word the high end implies a bright image
3 Histogram with two major peaks, called bimodal, implies an object that is in contrast with the background

Examples of the different types of histograms are shown in Figure (1-1).

Dr. Nidaa F. Hssan Image Processing

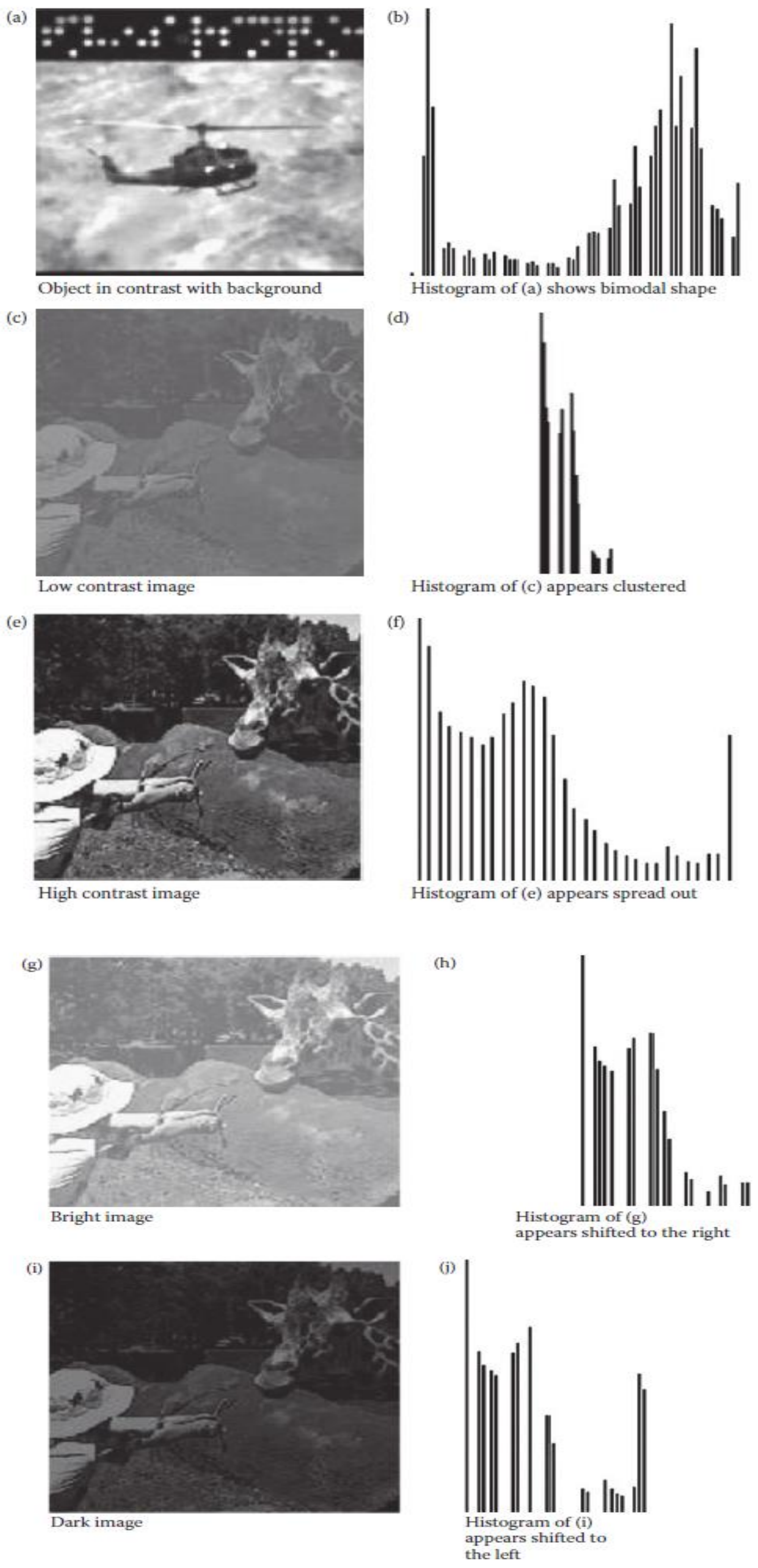
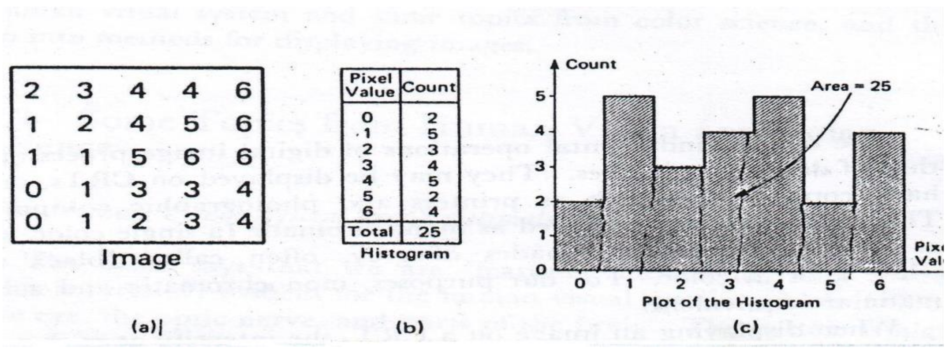


Figure (1-1): Different types of Histogram



2	3	4	4	6
1	2	4	5	6
1	1	5	6	6
0	1	3	3	4
0	1	2	3	4

Image

(a)

Pixel Value	Count
0	2
1	5
2	3
3	4
4	5
5	2
6	4
Total	25

Histogram

(b)

Figure (1-2): Sub image and its Histogram

1.2 <u>Histogram Modifications</u>

The gray level histogram of an image is the distribution of the gray level in an image. The histogram can be modified by mapping functions, which will stretch, shrink (compress), or slide the histogram. Figure (1-3) illustrates a graphical representation of histogram stretch, shrink and slide.

Dr. Nidaa F. Hassan

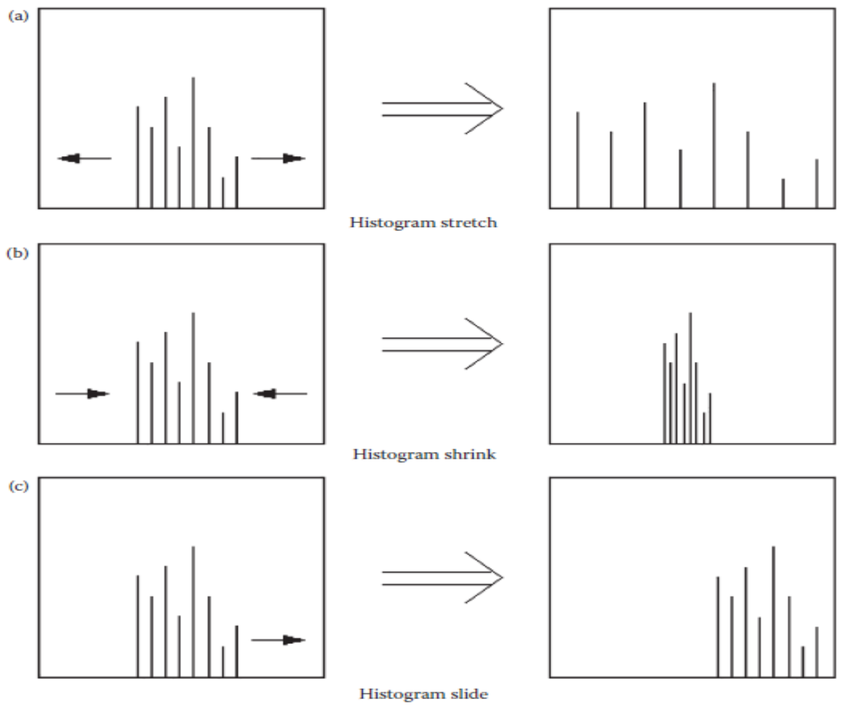


Figure (1-3): Histogram Modifications.

- The histogram can be modified by mapping functions, which are :

1. **Stretch Mapping Function.**

The mapping function for histogram stretch can be found by the following equation:

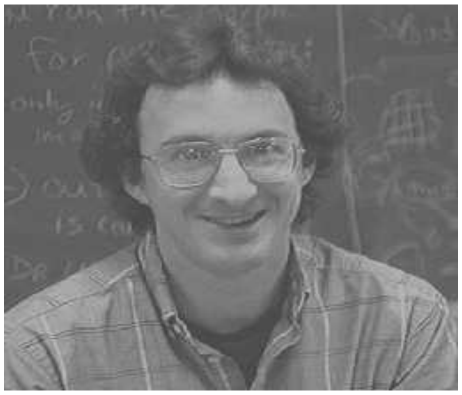
$$\text{Stretch (I (r, c))} = \left[\frac{I(r,c) - I(r,c)_{\min}}{I(r,c)_{\max} - I(r,c)_{\min}} \right] [MAX\text{-}MIN] + MIN.$$

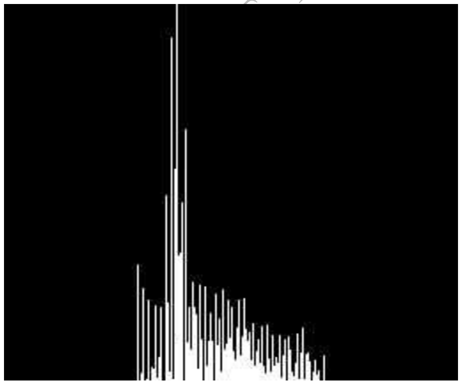
Where, I(r, c) $_{\max}$ is the largest gray- level in the image I(r, c).

I(r, c) $_{\min}$ is the smallest gray-level in the image I(r, c).

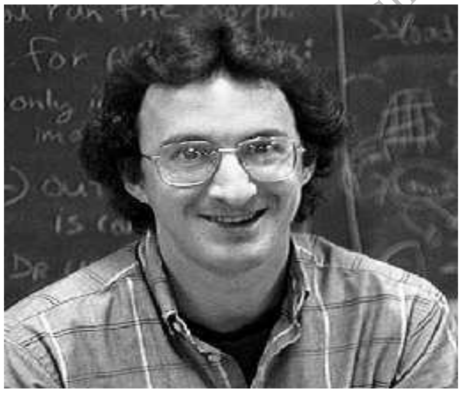
MAX and MIN correspond to the maximum and minimum gray–level values possible (for an 8-bit image these are 255 and 0).

Hassan


a. Low-contrast image


b. Histogram of low-contrast image

ing


c. Image after histogram stretching

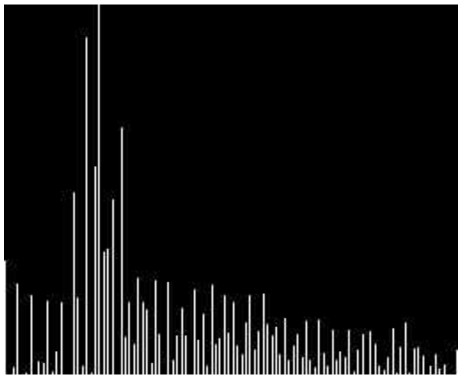

d. Histogram of image after stretching

Figure (1-4): Histogram Stretching.

This equation will take an image and stretch the histogram across the entire gray-level range which has the effect of increasing the contrast of a low contrast image (see Figure (1-5) of histogram stretching).

Most of the pixel values in an image fall within a small range, but a few outlines force the histogram to span the entire range, a pure histogram stretch will not improve the image. In this case, it is useful to allow a small proceeding of the pixel values to be aliped at the low and high end of the range (for an 8-bit image this means truncating at 0 and 255). See figure (2.20) of a stretched and clipped histogram).

Original Image

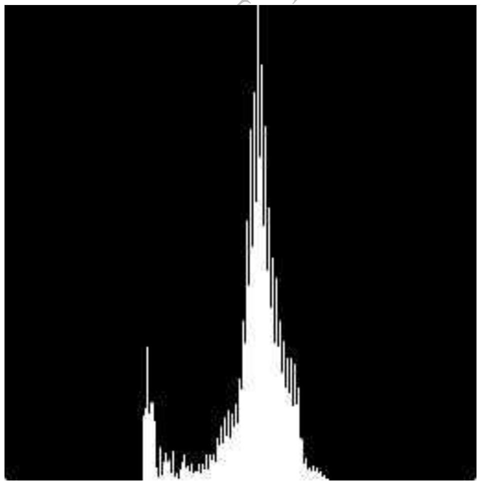
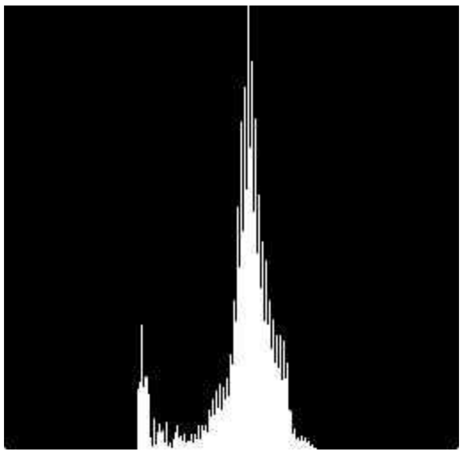
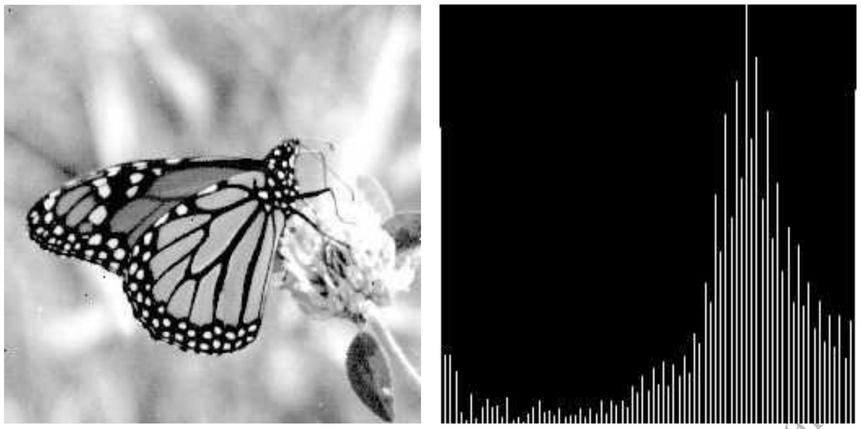

Histogram of the original image

Image after histogram stretching without clipping


Histogram of the image



Figurer (1-5): Histogram Stretching (Clipping).

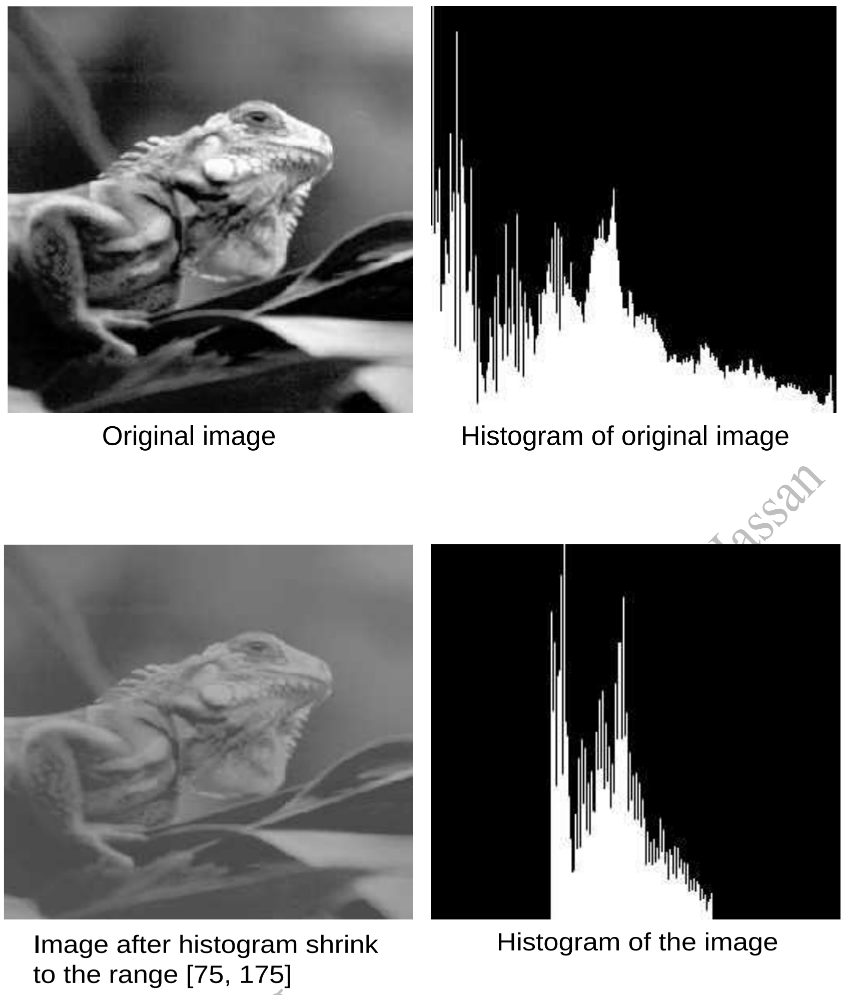
2. Shrink Mapping Function

The opposite of a histogram stretch is a histogram shrink, which will decrease image contrast by compressing the gray levels. The mapping function for a histogram shrinking can be found by the following equation:

$$\text{Shrink}\,(I(r,c)) = \left[\frac{Shrink_{\max} - Shrink_{\min}}{I(r,c)_{\max} - I(r,c)_{\min}}\right] [I(r,c) - I(r,c)_{min}] + Shrink_{min}$$

Shrink $_{\max}$ and shrink $_{\min}$ correspond to the maximum and minimum desired in the compressed histogram. In general, this process produces an image of reduced contrast and may not seem to be useful for an image enhancement (see figure (5-6) of shrink histogram).

Image Processing Dr. Nidaa F. Hassan



Figurer (1-6): Histogram Shrinking.

Image Processing ... Hassan

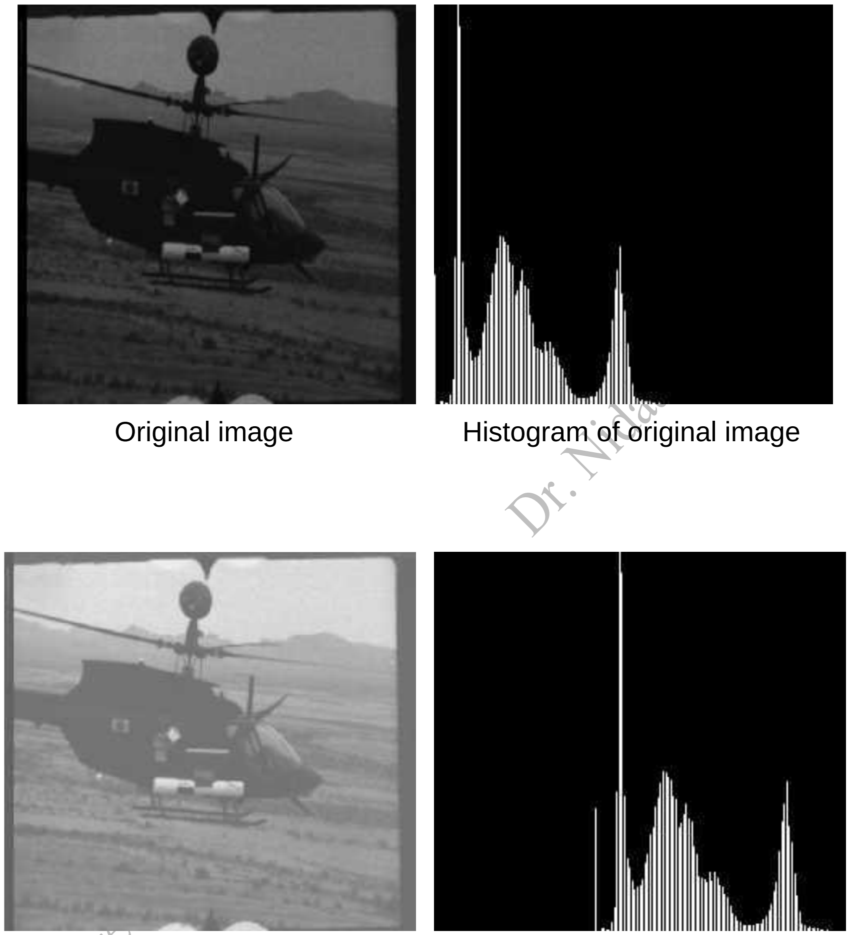
3. Slide technique

The histogram slide techniques can be used to make an image either darker or lighter but retain the relationship between gray-level values. This can be accomplished by simply adding or subtracting a fixed number for all the gray-level values, as follows:

$$\text{Slide}\,(I(r, c)) = I\,(r, c) + \text{OFFSET}.$$

Where OFFSET values are the amount to slide the histogram.

In this equation, a positive OFFSET value will increase the overall brightness; whereas a negative OFFSET will create a darker image, figure (1-7) shows histogram sliding.



Figurer (1-7): Histogram Sliding.

1.3 <u>Histogram Equalization</u>

Histogram Equalization is a popular technique is used for improving the appearance of a poor image. Its function is similar to that of a histogram stretch but often provides more visually pleasing results across a wide range of images.

Histogram equalization is a technique where the histogram of the resultant image is as flat as possible (with histogram stretching the overall shape of the histogram remains the same).

The results in a histogram with a mountain grouped closely together to "spreading or flatting histogram makes the dark pixels appear darker and the light pixels appear lighter (the keyword is "appear" the dark pixels in a photograph cannot by any darker. If, however, the pixels that are only slightly lighter become much lighter, then the dark pixels will appear darker).

Dr. Nidaa F . Hassan

The histogram equalization process for digital images consists of four steps:

1. Find the running sum of the histogram values
2. Normalize the values from step1 by dividing by the total number of pixels.
3. Multiply the values from step2 by the maximum gray level value and round.
4. Map the gray-level values to the results from step 3, using one-to-one correspondence. The following example will help to clarify this process.

Image Processing

Example:-

We have an image with 3 bit /pixel, so the possible range of values is 0 to 7. We have an image with the following histogram:

Gray-level value	0	1	2	3	4	5	6	7
No of Pixel Histogram value	10	8	9	2	14	1	5	2

Step 1: Great a running sum of histogram values. This means that the first values is 10, the second is 10+8=18, next is 10+8+9=27, and soon. Here we get 10,18,29,43,44,49,51.

Step 2: Normalize by dividing by total number of pixels. The total number of pixels is 10+8+9+2+14+1+5+0=51.

Step 3 : Multiply these values by the maximum gray – level values in this case 7 , and then round the result to the closet integer. After this is done we obtain 1,2,4,4,6,6,7,7.

Step 4 : Map the original values to the results from step3 by a one –to-one correspondence.

Dr. Nidaa F. Hassan

The first three steps:

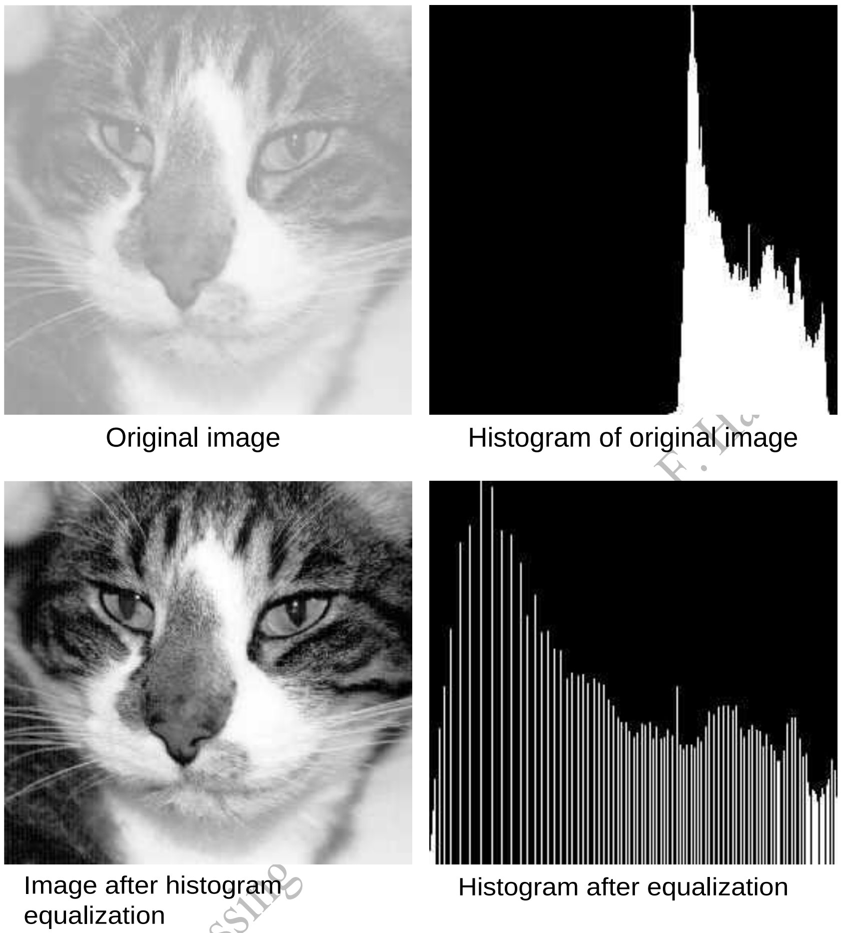
Gray-level	0	1	2	3	4	5	6	7
No. of Pixel	10	8	9	2	14	1	5	2
Run Sum	10	18	27	29	43	44	49	51
Normalized	10/51	18/51	27/51	29/51	43/51	44/51	49/51	51/51
Multiply by 7	1	2	4	4	6	6	7	7

Image Processing

The fourth step:

Old	0	1	2	3	4	5	6	7
New	1	2	4	4	6	6	7	7

All pixel in the original image with gray level 0 is set to 1, values of 1 are set to 2, 2 set to 4, 3 sets to 4, and so on (see figure (4-8)) histogram equalization, you can see the original histogram and the resulting histogram equalized histogram. Although the result is not flat, it is closer to being flat than the original.



Figurer (1-8): Histogram Equalization.

Image Processing

1.4 Histogram Features

The histogram features that we are considered are statically based features where the histogram is used as a model of the probability distribution of the gray levels. These statistical features provide us with information about the characteristic of the gray–level distribution for the image or sub-image. We define the **first-order histogram probability P(a)** as :

$$P(g) = \frac{N(g)}{M} \text{ ………….. } \mathbf{(1)}$$

M is the number of pixels in the image or sub-image (if the entire image is under consideration then M= N^2 for N×N). N (g) is the number of pixels at gray level g. As with any probability distribution, all values for P (g) are less than or equal to 1, histogram probability is mean, standard deviation, energy, and entropy.

Dr. Nidaa F. Hassan

1. **Mean:** the mean is the average value, so it tells us something about the general brightness of the image. A bright image will have a high mean, and a dark image will have a low mean. We will use L as the total number of gray levels available, so the gray levels range from 0 to L_1. For example, for typical 8-bit image data, L is 256 and ranges from 0 to 255. We can define the mean as follows:

Image Processing

$$\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_{r}\sum_{c} \frac{I(r,c)}{M} \text{.............}(2)$$

If we use the second form of the equation, we sum over the rows and columns corresponding to the pixels in the image or sub-image under consideration.

2. **Standard deviation:** Standard deviation is also known as the square root of the variance, tell us something about the contrast. It describes the spread in the data, so a high contrast image will have a high variance, and a low–contrast image will have a low variance. It is defined as follows:

$$\sigma = \sqrt{\sum_{g=0}^{L-1} (g-\bar{g})^2 P(g)} \,\ldots\ldots\ldots\, (3)$$

3. **Energy:** The energy measure tell us something about how the gray level are distributed

$$\text{Energy} = \sum_{g=0}^{L-1} [P(g)]^2 \ldots\ldots\ldots\ldots (4)$$

The energy measure has a maximum value of 1 for an image with a constant value and gets increasingly smaller as the pixel values are distributed across more gray level values(remember that all the P(g) values are less than or equal to 1). The larger this value is, the easier it is to compress the image data. If the energy is high, it tells us that the number of gray levels in the image is few, that is, the distribution is concentrated in only a small number of different gray levels.

Dr. Nidaa F. Hasson

Image Processing

4. **Entropy:** the entropy is a measure that tells us how many bits we need to code the image data and given by :

$$Entropy = -\sum_{g=0}^{L-1} P(g)\log_2[P(g)] \ldots\ldots\ldots (5)$$

As the pixel values in the image are distributed among more gray levels, the entropy increases. This measure tends to vary inversely with the energy.

Chapter Two
Image segmentation

2.1 Introduction

Image Processing and Computer Vision are concerned with developing algorithms for analyzing the contents of an image. One such approach for understanding an image is Pattern Recognition. Given a digitized image containing several objects the pattern recognition process consists of three major phases: Image Segmentation, Feature Extraction, and Classification. Where image segmentation is the process of isolation of objects, in feature extraction, objects are measured according to some quantifiable property of an object and classification tells which class each object belongs to. Our topic of focus is Image segmentation.

Image Processing Nidaa F Hasan

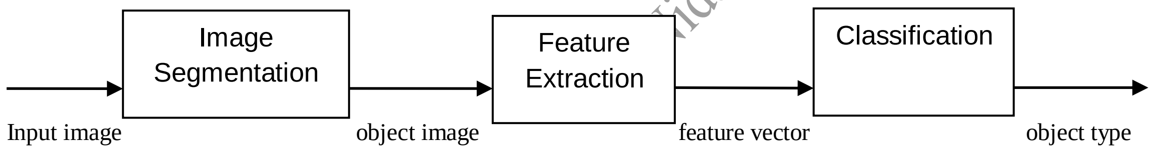


Figure (2.1) : pattern recognition process

<u>Image segmentation</u> is the process of partitioning the digital image into multiple regions that can be associated with the properties of one or more criteria.

Regions are elementary picture elements in a segmented image, formed as an aggregation تجميع of pixels, by two types of properties (relations):

-Their internal properties, like **<u>color, texture, shape,</u>** etc. help to identify regions.

-Their external relations, like adjacency, inclusion, and similarity of properties, are used to build groups of regions having a particular meaning in a more abstract context.

A person in a picture cannot be identified by searching for a region with a single color or texture; it is the constellation (group, collection) of projected surface patches (بقع) that allow the recognition of a person in a picture. The union of regions forming the group is again a region with both internal and external properties and relations. The first question that comes in mind is **how these natural groupings are found?.**

In other words what makes pixels in a partition to be more like one another than pixels in other segments? This observation pours down into two issues:

- **How to measure the similarity between pixels and regions, and**
- **How to evaluate a partitioning of the pixels into segments.**

Image segmentation methods will look for that either **have some measure of homogeneity within themselves or have some measure of contrast with the objects on their border**. Most image segmentation algorithms are modification, extension, or combinations of these two basic concepts.

The homogeneity (متجانس) and contrast measures can include features such as gray level, color, and texture.

In a mathematical sense the **segmentation of the image *I***, which is a set of pixels, is the partition of *I* into *n* disjoint sets *R*1,*R*2, . . . , *Rn,* called segments or regions, such that the union of all regions equals *I*.

$$I = R_1 \cup R_2 \cup \ldots \cup R_n \quad \ldots\ldots\ldots\ldots (1)$$

There is **no theory** of image segmentation. No single standard of image segmentation has emerged, rather is **a collection of (unplanned, informal) methods** that have received some degree of popularity.

Image segmentation techniques can be divided into four main categories:

1. Region-based segmentation
2. Clustering method.
3. Boundary detection.
4. Combined approaches.

Dr. Nidaa F. Hassan Image Processing

1- Region-based segmentation

Region-based methods focus attention on finding the regions in the image in comparison to the pixel. Let R be the entire image region, we may view segmentation as a process that partitions R into n sub-regions, R_1, R2, …, R_n such that

a) $_{i=1}\bigcup^{n} R_i=R$

b) R_i is a connected region , i = 1,2,3…..,n.

c) $R_i \cap Rj = ø$ for all, i and j and $i \neq j$

d) $P(R_i)$ = TRUE for i =1,2,3…..,n.

e) $P (R_i \cup R_j)$ =false for $i \neq j$.

Here, $P (R_i)$ is the logical predicate defined over the points in set R_i and ø is the null set.

Dr. Nidaa F.Hassan

- Condition (a) indicates segmentation should be complete; that is every pixel must be in a region.
- Condition (b) requires that points in a region must be connected in some predefined connectivity.
- Condition (c) indicates that the regions must be disjoint.
- Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region.
- Finally, condition (e) indicates that regions R_i and R_j are different in the sense of predicate P.

Image Processing

There are several methods to do region segmentation. Region-based segmentation methods segment the image into regions by operating principally in the re-based image. Some of the techniques used are:

- Local: in which small areas of the image are processed at a time.
- Global: in which the entire image is considered during processing.
- Methods that combine local and global techniques, such as split and merge.

Discussion of few these methods follows:

1-1 Region Growing and Shrinking.

With this approach, one begins with a set of seed points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the initial seed. These regions can be small neighborhoods or single pixels. The properties that distinguish the pixels inside the different objects might include average gray level, texture, or color information. Next all boundaries between the regions are examined. A measure of boundaries strength is computed using differences of the average properties of the adjacent regions. A given boundary is strong if the properties differ on either side of the boundary significantly and it is weak if they do not. Strong boundaries are allowed to stand and weak boundaries are dissolved and adjacent regions merged. The process is iterated by alternately re-computing the object membership properties for the enlarged regions and then dissolving weak boundaries. The region merging process is continued until there are no more boundaries that are weak enough to dissolve. Then segmentation is said to be completed.

Dr. Nidaa F.Hassan Image Processing

For example, take an X-ray image of a weld containing several tracks and porosities. We are segmenting to see the weld failures. The first thing to do here is to determine the initial seed points. In this application, it is known that pixels of defective welds tend to have the maximum allowable digital values. Based on this information select starting points as pixels having values of 255.

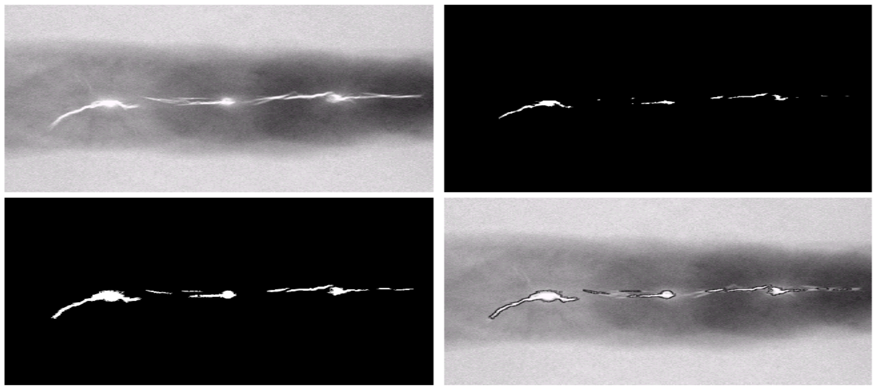


Figure (2.2)
(a) Image showing defective welds
(b) Seed points
(c) Results of region growing
(d) Boundaries of segmented defective welds (in black)

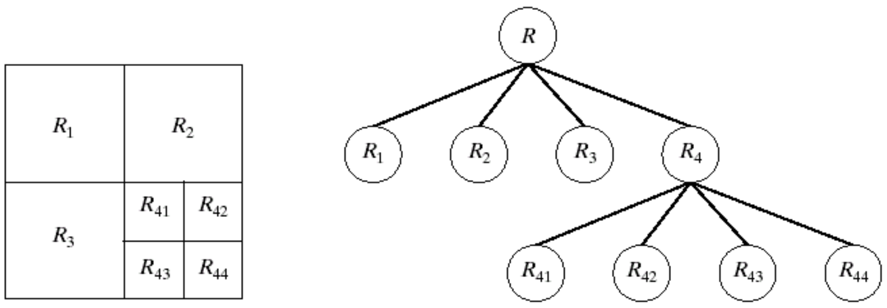
1-2 Split and Merge Techniques

In general, the split and merge techniques proceed as follows:

1. Define a homogeneity test. This involves defining a homogeneity measure, which may incorporate brightness, color, texture, or other application-specific information, and determining a criterion the region must need to pass the homogeneity test.
2. Split the image into equally sized regions.
3. Calculate the homogeneity measure for each region.
4. If the homogeneity test is passed for the region then a merge is attempted with its neighbor(s). If the criterion is not met the region is split.
5. Continue this process until all regions pass the homogeneity test.
    - Split and merge image segmentation techniques are based on the quadtree data representation. That is a square image segment is broken into four quadrants if the original image is non-uniform in the attribute. If the four neighboring squares are uniform then they are replaced with a single attribute. The basic split and merge process tends to produce rather blocky segments because of the rule that square blocks are either split or merged.

Dr. Nidaa F.Hassan
Image Processing

Figure (2.3)
Partitioned image
Corresponding quadtree



2- Clustering Techniques:

Clustering techniques are image segmentation methods by which individual elements are placed into groups. These groups are based on some measures of similarity within the group.

The major difference between these techniques and the region-growing techniques is that domains other than RC-based image space (the spatial domain) may be considered as the primary domain for clustering. Some of these other domains include color space, histogram spaces, or complex feature spaces.

Recursive region splitting is a clustering method that has become a standard technique. This method has become a standard technique (this method uses thresholding of histogram techniques to segment the image). This algorithm proceeds as follows:

1. Consider the entire image as one region and compute histograms for each component of interest (for example, red, green, and blue for color image).
2. Apply a peak finding test to each histogram. Select the best peak and put thresholds on either side of the peak. Segment the image into two regions based on this peak.
3. Smooth the binary threshold image so that only a single connected sub-region is left.

Image Processing Dr. Nidaa F.Hassan

4. Repeat steps 1-3 for each region until no new sub-regions can be created that is no histogram has a significant peak.

3- <u>Boundary Detection</u>:

Is performed by finding the boundaries between objects. Thus indirectly defining the objects, this method is usually begun by marking points that may be a part of an edge. These points are then merged into line segments then merged into object boundaries. The edge detectors previously described are used to mark points of rapid change, thus indicating the possibility of an edge. These edge points represent local discontinuities in specific features such as brightness, color, or texture.

- After the edge detection operation has been performed the next step is to threshold the result.

One method to do this is to consider the histogram of the edge detection results, looking for the best valley as shown in the following figures.

Dr. Nidaa F.Hassan Image Processing

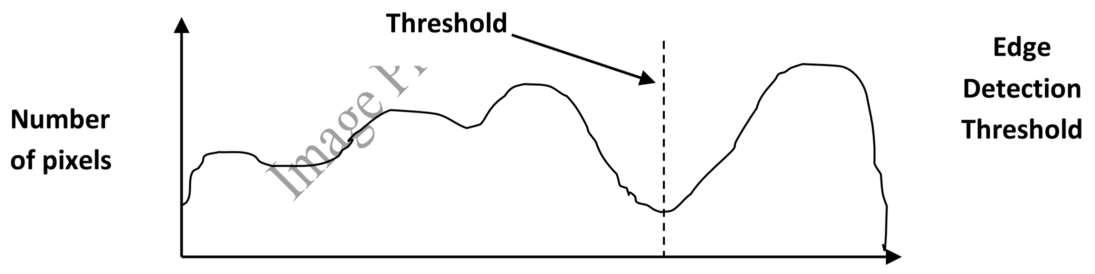


Figure (2.4): Pixel value after edge detection.

Often the histogram of an image that has been operated by an edge detector is unimodal (one peak), so it may difficult to find a good valley. This method works best with a bimodal histogram. Another method that provides a reasonable result is to use the average value for the threshold with a very noisy image, a good rule is to use 10-20% of the peak value as a threshold.

4- <u>Combined Technique</u>:

Image segmentation methods may be a combination of region growing methods, clustering methods, and boundary detection. Optimal image segmentation is likely to be achieved by focusing on the application and on how the different methods can be used, single or in combination to achieve the desired result.

Image Processing Dr. Nidaa F.Hassan

Chapter Three
Image Compression

3.1 Introduction

Image compression involves reducing the size of the image data file, while is retaining necessary information, the reduced file is called the compressed file and is used to reconstruct the image, resulting in the decompressed image. The original image, before any compression is performed, is called the uncompressed image file. The ratio of the original, uncompressed image file and the compressed file is referred to as the compression ratio.

$$Compression\ Ratio = \frac{Uncompressed\ File\ Size}{Compressed\ File\ Size} = \frac{Size_U}{Size_C}$$

It is often written as Size $_U$: Size $_C$

For example the original image is 256×256 pixel, single-band (gray-scale), 8-bit per pixel. This file is 65,536 bytes (64K). After compression, the image file is 6,554 bytes. The compression ratio is:

Size $_U$: Size $_C$ = $\frac{65536}{6554} \cong 9.999 \cong 10$ this can be written as 10:1

This is called a "10 to 1" compression or a "10 times compression", or it can be stated as "compressing the image to 1/10 original size.

Another way to state the compression is to use the terminology of bits per pixel. For an N×N image

$$\textbf{Bit per Pixel} = \frac{Number\ \ of\ \ Bites}{Number\ \ of\ \ Pixels} = \frac{8(Number\ \ of\ \ Bytes)}{N \times N}$$

Example: using the preceding example, with a compression ratio of 65536/6554 bytes. We want to express this as bits per pixel. This is done by first finding the number of pixels in the image.

256×256 = 65,536

We then find the number of bits in the compressed image file

6554× (8 bit/byte) =52432 bits.

Now, we can find the bits per pixel by taking the ration

52432/65536=0.8 bit/ Pixel.

The reduction in file size is necessary to meet

1. The bandwidth (كميـة البيانـات) requirements for many transmission systems.
2. The storage requirements in the computer database.

The amount of data required for digital images is enormous. For example, a single 512×512, 8-bit image requires 2,097,152 bits for storage. If we wanted to transmission this image over the World Wide Web, it would probably take minutes for transmission –too long for most people to wait.

512 ×512×8= 2,097,152.

Example

To transmit a digitized color scanned at 3,000×2,000 pixels, and 24 bits, via modem at 28.8(kilobits/second), it would take about

$$\frac{(3000\times 2000\ \text{pixels})(24\ \ \text{bits/pixel s})}{(28.8\times 1024\ \text{bits}\ /\text{second})} \cong 4883\ \text{Second}$$

$$= 81\ \text{minutes}\ .$$

Couple this result with transmitting multiple image or motion images, and the necessity of image compression can be appreciated.

The key to a successful compression schema comes with the second part of the definition –retaining necessary information. To understand this we must differentiate between data and information.

For digital images, data refer to pixel gray-level values correspond to the brightness of a pixel at a point in space. Information is the interpretation of the data in a meaningful way. Data are used to convey information, much like the way the alphabet is used to convey information via words. Information is an elusive concept, it can be application-specific. For example, in a binary image that contains text only, the necessary information may only involve the text being readable, whereas for a medical image the necessary information may be every minute detail in the original image.

There are two primary types of images compression methods and they are:

1. Lossless Compression
2. Lossy Compression.

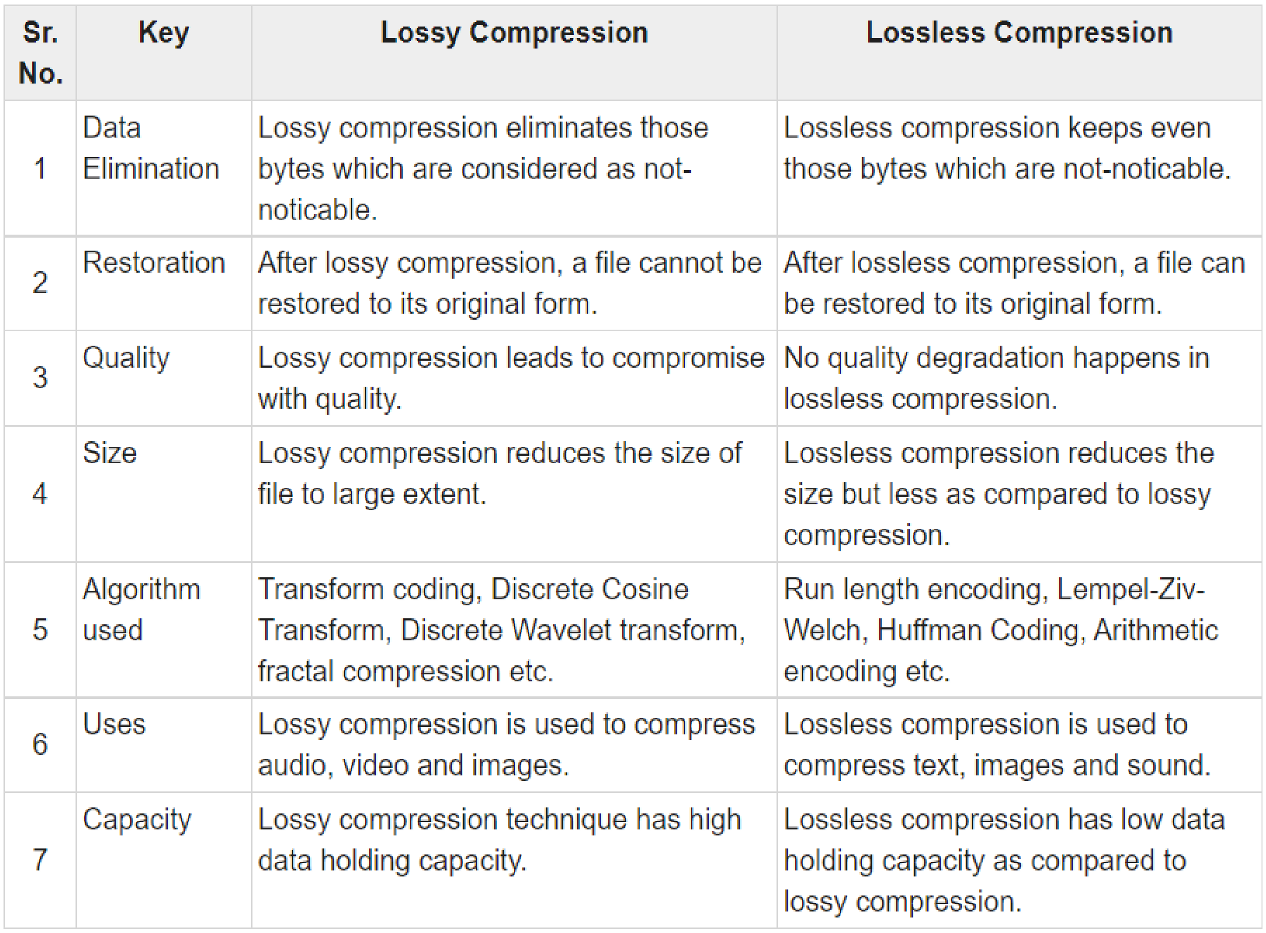
1. Lossless Compression

This compression is called lossless because no data are lost, and the original image can be recreated exactly from the compressed data. For simple images such as text-only images.

2. Lossy Compression.

These compression methods are called Lossy because they allow a loss in actual image data, so the original uncompressed image cannot be created exactly from the compressed file. For complex images, these techniques can achieve compression ratios of 100 0r 200 and still retain in high – quality visual information. For simple image or lower-quality results compression ratios as high as 100 to 200 can be attained. Table (3.1) shows differences between lossy and lossless compression

Table (3.1) : Differences between lossy and lossless compression



Sr. No.	Key	Lossy Compression	Lossless Compression
1	Data Elimination	Lossy compression eliminates those bytes which are considered as not-noticable.	Lossless compression keeps even those bytes which are not-noticable.
2	Restoration	After lossy compression, a file cannot be restored to its original form.	After lossless compression, a file can be restored to its original form.
3	Quality	Lossy compression leads to compromise with quality.	No quality degradation happens in lossless compression.
4	Size	Lossy compression reduces the size of file to large extent.	Lossless compression reduces the size but less as compared to lossy compression.
5	Algorithm used	Transform coding, Discrete Cosine Transform, Discrete Wavelet transform, fractal compression etc.	Run length encoding, Lempel-Ziv-Welch, Huffman Coding, Arithmetic encoding etc.
6	Uses	Lossy compression is used to compress audio, video and images.	Lossless compression is used to compress text, images and sound.
7	Capacity	Lossy compression technique has high data holding capacity.	Lossless compression has low data holding capacity as compared to lossy compression.

3.2 <u>Compression System Model</u>

The compression system model consists of two parts: Compressor and Decompressor as shown in figure (3.1)

Compressor: consists of preprocessing stage and encoding stage, as shown in figure (3.2)

Decompressor: consists of decoding stage followed by a post-processing stage, as shown in figure (3.3)

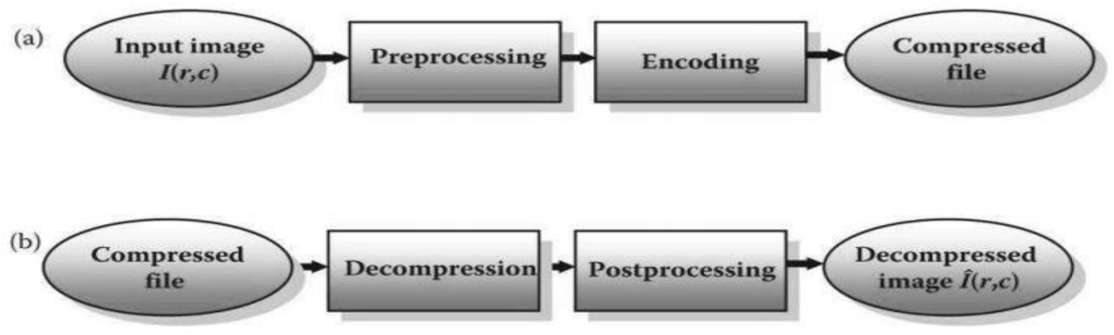


Figure (3.1): Compression System Model, (a) Compression, (b) decompression..

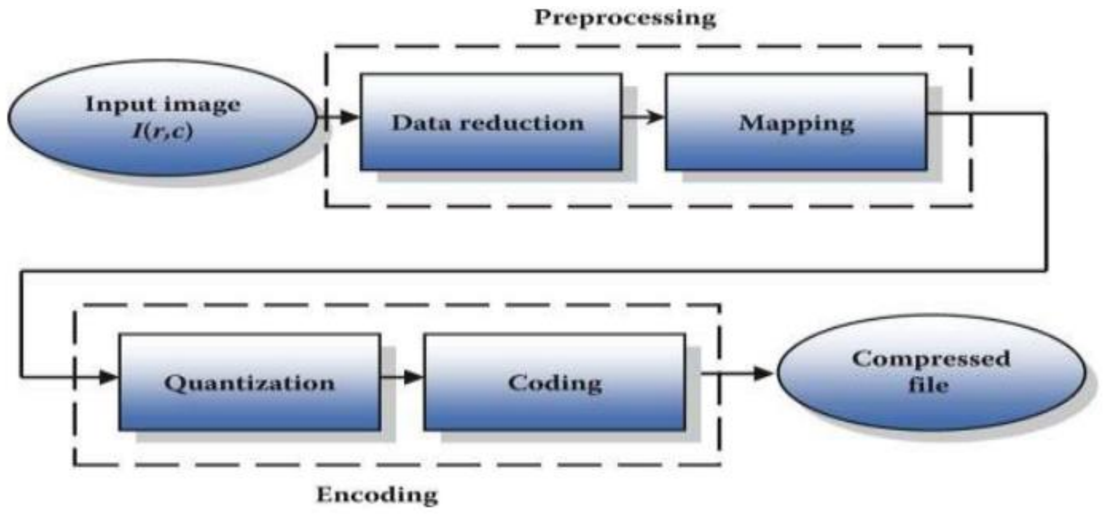


Figure (3.2): The compressor.

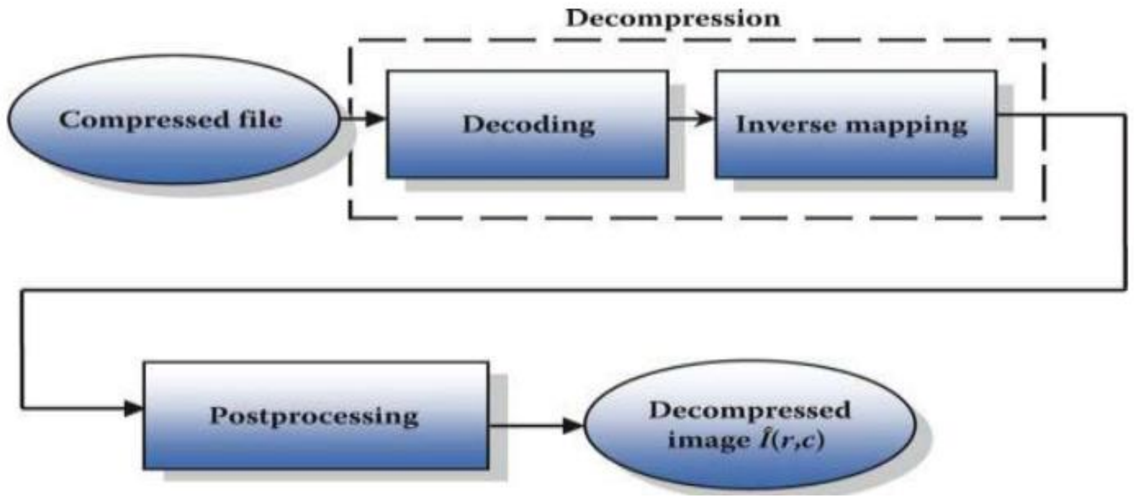


Figure (3.3): The Decompressor.

Before encoding, preprocessing is performed to prepare the image for the encoding process, and consists of any number of application-specific operations.

The compressor can be further broken down into stages.

A. Preprocessing

- **data reduction:** the image data can be reduced by gray level and/or spatial quantization.
- **mapping process:** which maps the original image data into another mathematical space where it is easier to compress the data.

B. Encoding process: it is the **quantization** stage, which takes the potentially continuous data from the mapping stage and puts it in discrete form. The final stage of encoding involves coding the resulting data, which maps the discrete data from the quantizer onto a code in an optimal manner.

- **Quantization** may be necessary to convert the data into digital form (BYTE data type). This is because many of these mapping methods will result in floating point data that requires multiple bytes for representation—not very efficient if our goal is data reduction. There are two ways to do the quantization:

  – .
  – **Nonuniform quantization:** These quantization bins are not all of equal width. The coding stage of any image compression algorithm is very important. The coder provides a one-to-one mapping, each input is mapped to a unique output by the coder, so it is a reversible process. The code can be an equal length code, where all the code words are the same size, or an unequal length code with variable length code words

After the compressed file has been decoded, post-processing can be performed to eliminate some of the undesirable artifacts brought about by the compression process. Often, many practical compression algorithms are a combination of many different individual compression techniques.

3.3 Lossless Compression Method

Lossless compression methods are necessary for some imaging applications. For example with medical images, the law requires that any archived medical images are stored without any data loss. Many of the lossless techniques were developed for non-image data and, consequently, are not optimal for image compression. In general, the lossless techniques alone provide compression in the range of only a 10% reduction in file size. However lossless compression techniques may be used for both preprocessing and postprocessing in image compression algorithms. Additionally, for a simple image, the lossless techniques can provide subnational compression.

An important concept here is the idea of measuring the average information in an image, referred to as entropy. The entropy for the N×N image can be calculated by this equation.

$$Entropy = -\sum_{i=0}^{L-1} P_i \log_2(P_i) \qquad \text{(In bits per pixel)}$$

Where

P_i = The probability of the ith gray level $\frac{n_k}{N^2}$

n_k= the total number of pixels with gray value k.

L= the total number of gray levels (e.g. 256 for 8-bits)

Example

Let L=8, meaning that there are 3 bits/ pixel in the original image. Let that number of the pixel at each gray level value is equal (they have the same probability) that is:

$$P_0 = P_1 = P_2 = \cdots\cdots = P_7 = \frac{1}{8}$$

Now, we can calculate the entropy as follows:

$$Entropy = -\sum_{i=0}^{7} P_i \log(P_i)$$

$$= -\sum_{i=0}^{7} \frac{1}{8} \log_2(\frac{1}{8}) = 3$$

This tells us that the theoretical minimum for lossless coding for this image is 8 bit/pixel

Note/ $\text{Log}_2(x)$ can be found by taking $\log_{10}$ and multiplying by 3.322

Example

Let L = 8, thus we have a natural code with 3 bits per pixel in the original image. Now let's say that the entire image has a gray level of 2, so

$$\boldsymbol{p2 = 1, \text{ and } p0 = p1 = p3 = p4 = p5 = p6 = p7 = 0}$$

and the entropy is

$$Entropy = -\sum_{i=0}^{L-1} P_i \log_2(P_i) = -\ (1) \log_2(1)+0+\ldots.+0=0$$

This tells us the theoretical minimum for coding this image is 0 bits per pixel. Why is this? Because the gray-level value is known to be 2. To code the entire image we need only one value, this is called a certain event, it has a probability of 1.

There are many methods used to compress image without any loss in data, some of these methods are:

A. Run Length Coding
B. Huffman Coding

A. Run-length coding (RLC)

Run-length coding **(RLC)** is an image compression method that works by counting the number of adjacent pixels with the same gray-level value. This count, called the ***run-length***, is then encoded and stored. There are several methods of run-length encoding: basic methods used primarily for binary (two-valued) images and extended versions for gray-scale images.

Basic RLC is used primarily for binary images but can work with complex images that have been preprocessed by thresholding to reduce the number of gray levels to two. There are various ways to implement basic RLC, and the first step is to define the required parameters. We can either use

- Horizontal RLC, counting along the rows, or
- Vertical RLC, counting along with the columns.

In basic horizontal RLC, the number of bits used for the encoding depends on the number of pixels in a row. If the row has 2^n pixels, then the required number of bits is n, so that a run that is the length of the entire row can be encoded. So:

A 256 × 256 image requires 8-bits, since $2^8 = 256$.

A 512 × 512 image requires 9-bits, since $2^9 = 512$.

The next step is to define a convention for the first RLC number in a row whether it represents a run of 0s or 1s? Defining the convention for the first RLC number to represent 0s, the following example explains RLC:

<u>Example:</u> The image is an 8 × 8 binary image, which requires 3 bits for each run-length coded word. In the actual image file are stored 1s and 0s, although upon display the 1s become 255 (white) and the 0s are 0 (black). To apply RLC to this image, using horizontal RLC:

0	0	0	0	0	0	0	0	8
1	1	1	1	0	0	0	0	0, 4, 4
0	1	1	0	0	0	0	0	1, 2, 5
0	1	1	1	1	1	0	0	1, 5, 2
0	1	1	1	0	0	1	0	1, 3, 2, 1, 1
0	0	1	0	0	1	1	0	2, 1, 2, 2, 0
1	1	1	1	0	1	0	0	0, 4, 1, 1, 2
0	0	0	0	0	0	0	0	8

The RLC numbers are
First row: 8
Second row: 0, 4, 4
Third row: 1, 2, 5
Fourth row: 1, 5, 2
Fifth row: 1, 3, 2, 1, 1
Sixth row: 2, 1, 2, 2, 1
Seventh row: 0, 4, 1, 1, 2
Eighth row: 8
Note that in the second and seventh rows, the first RLC number is 0, since we are using the convention that the first number corresponds to the number of zeros in a run.

<u>Example 2:</u> Given the following 8 × 8, 4-bit image:

10	10	10	10	10	10	10	10	10, 8
10	10	10	10	10	12	12	12	10, 5, 12, 3
10	10	10	10	10	12	12	12	10, 5, 12, 3
0	0	0	10	10	10	0	0	0, 3, 10, 3, 0, 2
5	5	5	0	0	0	0	0	5, 3, 0, 5,
5	5	5	10	10	9	9	10	5, 3, 10, 2, 9, 2, 10, 1
5	5	5	4	4	4	0	0	5, 3, 4, 3, 0, 2

0	0	0	0	0	0	0	0	0,8

The corresponding gray levels pairs are as follows:
First row: 10, 8
Second row: 10, 5 12, 3
Third row: 10, 5 12, 3
Fourth row: 0, 3 10, 3 0, 2
Fifth row: 5, 3 0, 5
Sixth row: 5, 3 10, 2 9, 2 10,1
Seventh row: 5, 3 4, 3 0, 2
Eighth row: 0, 8
These numbers are then stored in the RLC compressed file as
10,8,10,5,12,3,10,5,12,3,0,3,10,3,0,2,5,3,0,5,5,3,10,2,9,2,10,1,5,3,4,3,0,2,0,8

B. Huffman Coding

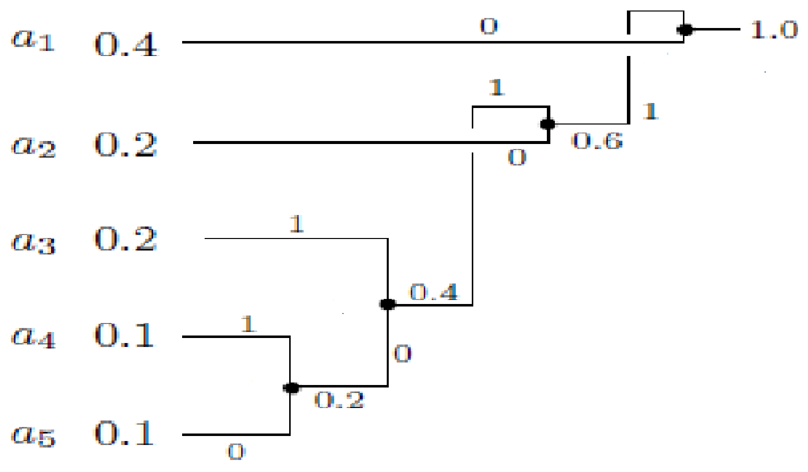
The Huffman code, developed by D. Huffman in 1952, is a minimum length code. This means that given the statistical distribution of the gray levels (the histogram), the Huffman algorithm will generate a code that is as close as possible to the minimum bound, the entropy. This method results in an **unequal (or variable) length code**, where the size of the code words can vary. For complex images, Huffman coding alone will typically reduce the file by 10–50% (1.1:1–1.5:1), but this ratio can be improved to 2:1 or 3:1 by preprocessing for irrelevant information removal.

The Huffman algorithm can be described in five steps:

1. Find the gray-level probabilities for the image by finding the histogram.
2. Order the input probabilities (histogram magnitudes) from smallest to largest.
3. Combine the smallest two by addition.
4. GOTO Step 2, until only two probabilities, are left.
5. By working backward along the tree, generate code by alternating assignments of 0 and 1.

This procedure is best illustrated by an example.

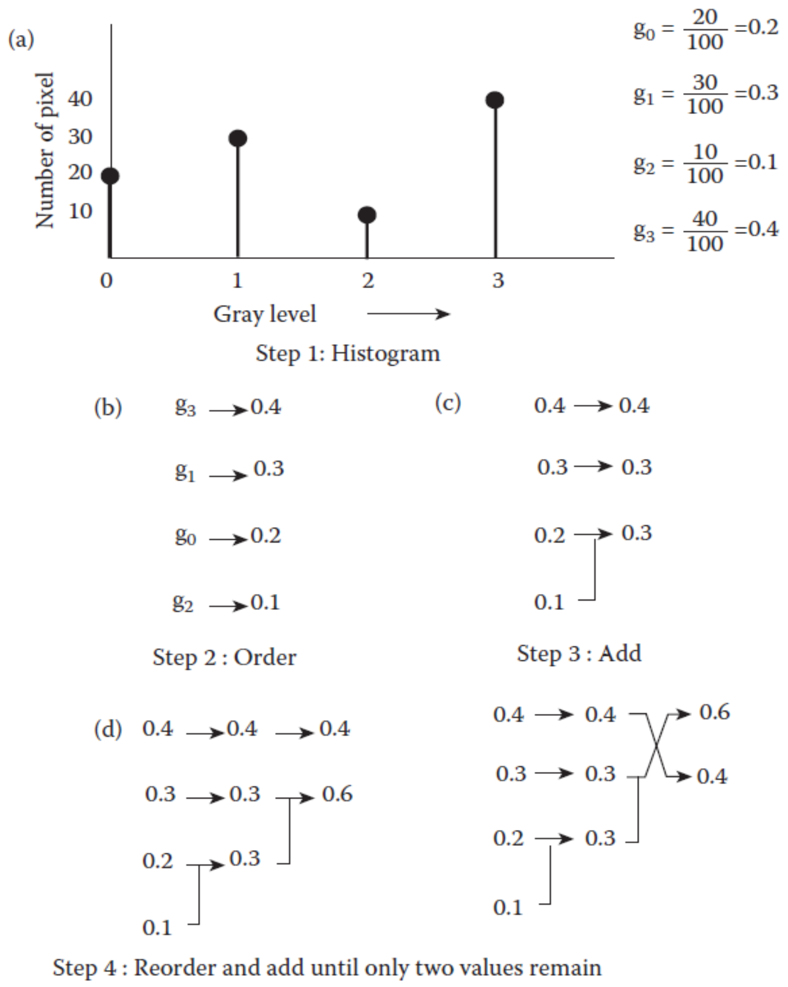
Example: The following five symbols a1, a2, a3, a4, and a5 have probabilities 0.4, 0.2, 0.2, 0.1, and 0.1 respectively. Compress it using Huffman coding?



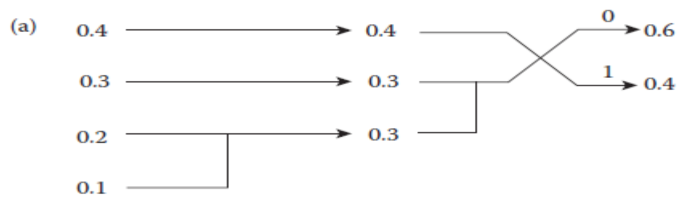
a1	0
a2	01
a3	111
a4	1101
a5	1100

Example:

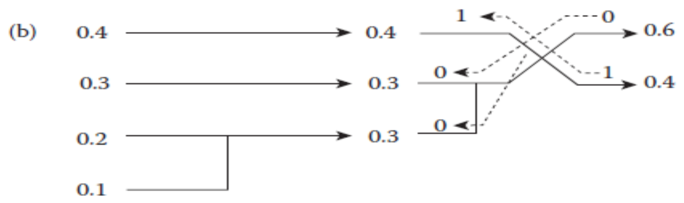
Assume that an image with 2-bits per pixel, giving four possible gray levels. The image is 10 rows by 10 columns. In Step 1 we find the histogram for the image. This is shown in Figure (6.2), where we see that gray level 0 has 20 pixels, gray level 1 has 30 pixels, gray level 2 has 10 pixels, and gray level 3 has 40 pixels with the value.



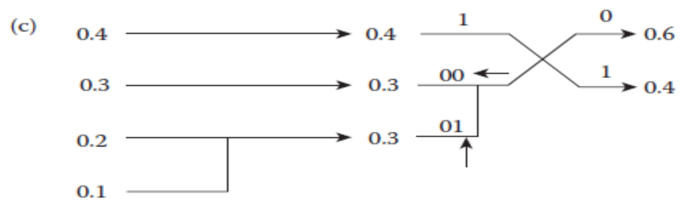
Step 5: generate code by alternating assignment of 0 and 1.



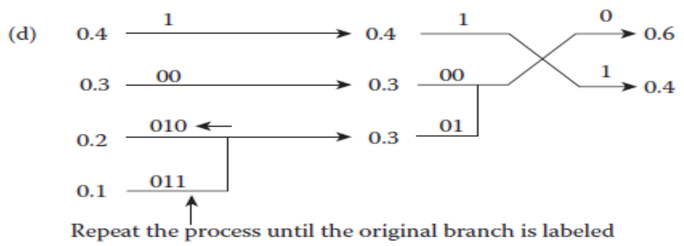
Assign 0 and 1 to the right–most probabilities



Bring 0 and 1 back along the tree



Append 0 and 1 to previously-added branches



Repeat the process until the original branch is labeled

Huffman Code for Example

Original Gray Level (Natural Code)	Probability	Huffman code
g_0: 00_2	0.2	010_2
g_1: 01_2	0.3	00_2
g_2: 10_2	0.1	011_2
g_3: 11_2	0.4	1_2

$$Entropy = -\sum_{i=0}^{3} p_i \log_2(p_i)$$

$$= -[(0.2)\log_2(0.2) + (0.3)\log_2(0.3) + (0.1)\log_2(0.1) + (0.4)\log_2(0.4)]$$

$$\approx 1.846 \text{ bits/pixel}$$

(Note: $\log_2(x)$ can be found by taking $\log_{10}(x)$ and multiplying by 3.322)

Chapter Four
Discrete Transform

4.1 Introduction

The transform maps image data into a different mathematical space via a transformation equation.

- However, the color transformations mapped data from one color space to another color space with a one-to-one correspondence between a pixel in the input and the output.
- Here, we are mapping the image data from the spatial المكانية (time) domain spectral الطيفية domain, where all the pixels in the input (spatial domain) contribute to each value in the output (frequency domain) as illustrated in the following figure:

Image Processing Dr. Nidaa F Hassan

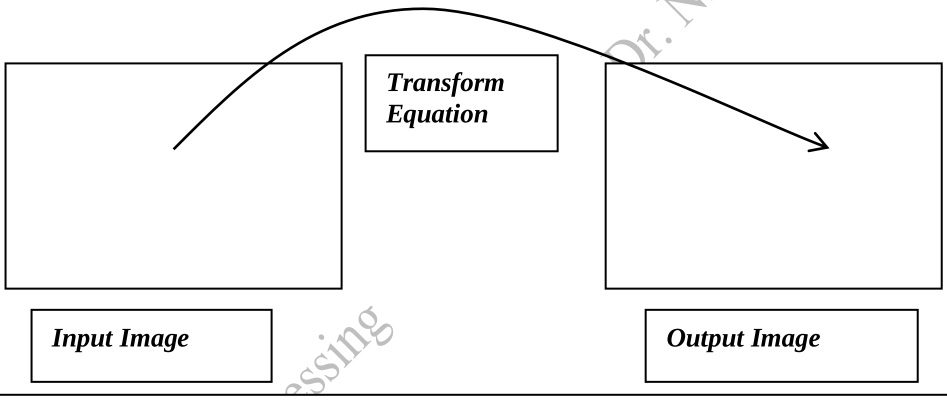


Figure (4.1): Color transform use a single –pixel to single –pixel mapping**.**

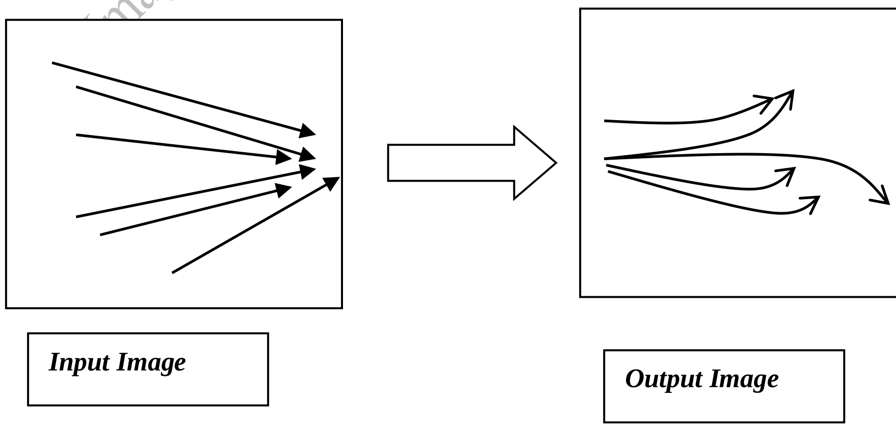


Figure (4.2): All pixels in the input image contribute to each value in the output image color transform use a single –pixel to single –pixel mapping**.**

- The most distinguished information is hidden in the frequency content.

Image transforms are designed to have two properties:

- To reduce image redundancy by reducing the sizes of most pixels and to identify the less important parts of the image by isolating the various frequencies of the image.

Image frequencies are important because of the following basic fact:

- Low frequencies correspond to the important image features, whereas high frequencies correspond to the details of the image, which are less important. Thus, when a transform isolates the various image frequencies, pixels that correspond to high frequencies can be quantized تكميم heavily بشكل كبير, whereas pixels that correspond to low frequencies should be quantized lightly or not at all. This is how a transform can compress an image very effectively by losing information, but only information associated with unimportant image details.

1. Fourier Transform (FT)

The Fourier transform is the most well-known, and the most widely used transform. It was developed by Baptiste Joseph Fourier (1768-1830) to explain the distribution of temperature and heat conduction التوصيل. Since that time the Fourier transform has found numerous uses, including vibration الاهتزاز analysis in mechanical engineering, and here in computer imaging. In mathematics, a Fourier series decomposes periodic functions or periodic signals into the sum of a (possibly infinite) set of simple oscillating تتأرجح functions, namely sine and cosine (or complex exponentials).

Dr. Nidaa F. Hassan
Image Processing

This transform allows for decomposition of an image into weighted sum of 2-d sinusoidalجيبية terms. Assuming an N×N image, the equation for the 2-D discrete Fourier

$$\mathbf{F(u,v)} = \frac{1}{N}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} I(r,c)e^{-j2\pi\frac{(ur+vc)}{N}}$$

The base of the natural logarithmic function e is about 2.71828; j, the imaginary coordinates for a complex number, equal $\sqrt{-1}$. The basis functions are sinusoidal in nature, as can be seen by Euler's identity:

$$e^{jx} = \cos x + j\sin x$$

So we can write the Fourier transform equation as

Nidaa F.Hassan

$$F(u,v) = \frac{1}{N}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} I(r,c)\left[\cos\left(\frac{2\pi}{N}(ur+vc)\right) - j\sin\left(\frac{2\pi}{N}(ur+vc)\right)\right]$$

In this case, F(*u*, *v*) is also complex, with the real part corresponding to the cosine terms and the imaginary part corresponding to the sine terms.

If we represent a complex spectral component

$$F(u,v)=R(u,v)+ j(u,v)$$

Image Processing

Where R(*u*,*v*) in the real part and I (*u*,*v*) is the imaginary part, then we can define the magnitude and phase of a complex spectral component us:

$$\text{Magnitude} = |F(u,v)| = \sqrt{[R(u,v)]^2 + [I(u,v)]^2}$$

$$\text{Phase}= \phi\ (u,v) = \tan^{-1}[\frac{I(u,v)}{R(u,v)}]$$

The magnitude of a sinusoid is simply its peak value, the phase, data contain information about where objects are in an image. After we perform the transform, if we want to get our original image back, we need to apply the inverse transform. The inverse Fourier transform is given by:

$$F^{-1}[F(u,v)] = I(r,c) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi\frac{(ur+vc)}{N}}$$

The following algorithm that illustrated the section of converting image (N×N) to frequency domain using Fourier transform:

1. **ALGORITHM: converting digital image to frequency domain using Fourier Transform**

Input: Input Image File (Picture1)

Output: Transformed Image (Picture 2)

```
Step1:  Picture2. Width = Picture1. Width
        Picture2. Height = Picture1.  Height
Step2:
For u = 0 to width of input image
    For v = 0 to height of input image
        Sum=0
    For r = 0 to width of input image
        For c = 0 to height of input image
Sum=Sum +I (r, c) * Cos [2* π /n * (u*r + v*c)] ─ J*sin [(2* π /n*(u*r + v*c)]
        Next c
    Next r
    F (u, v) = 1/n * Sum
    Next v
Next u
Step3: End
```

Dr. Nidaa F.Hassan

Image Processing

The figure below shows a gray image and its Fourier spectrum:

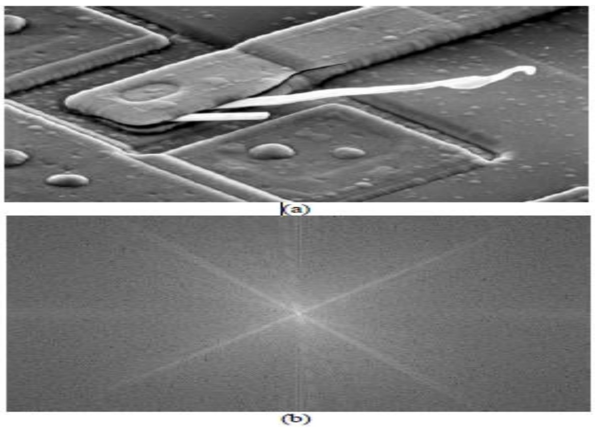


Figure (4.3) : (a) Gray image , (b) Fourier spectrum of (a).

2. Cosine Transform (CT)

The cosine transform, like Fourier transform, uses sinusoidal جيبية basis function. **The difference is that cosine basis functions are not complex, they use only cosine functions and not sine functions**. Assuming an N×N image, the discrete cosine transform equation is given by:

$$C(u,v) = \propto(u) \propto(v) \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} I(r,c) \cos\left[\frac{(2r+1)}{2N}\right] \cos\left[\frac{(2c+1)}{2N}\right]$$

$$\propto(u), \propto(v) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{for } u,v = 0 \\ \sqrt{\frac{2}{N}}, & u,v = 1,2,\ldots\ldots,N-1 \end{cases}$$

Dr. Nidaa F.Hassan

Because this transform uses only the cosine function, it can be calculated using on real arithmetic (not complex).

The inverse cosine transform is given by:

$$C^{-1}[C(u,v)] = I(r,c) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2r+1)u\pi}{2N}\right]\cos\left[\frac{(2c+1)v\pi}{2N}\right]$$

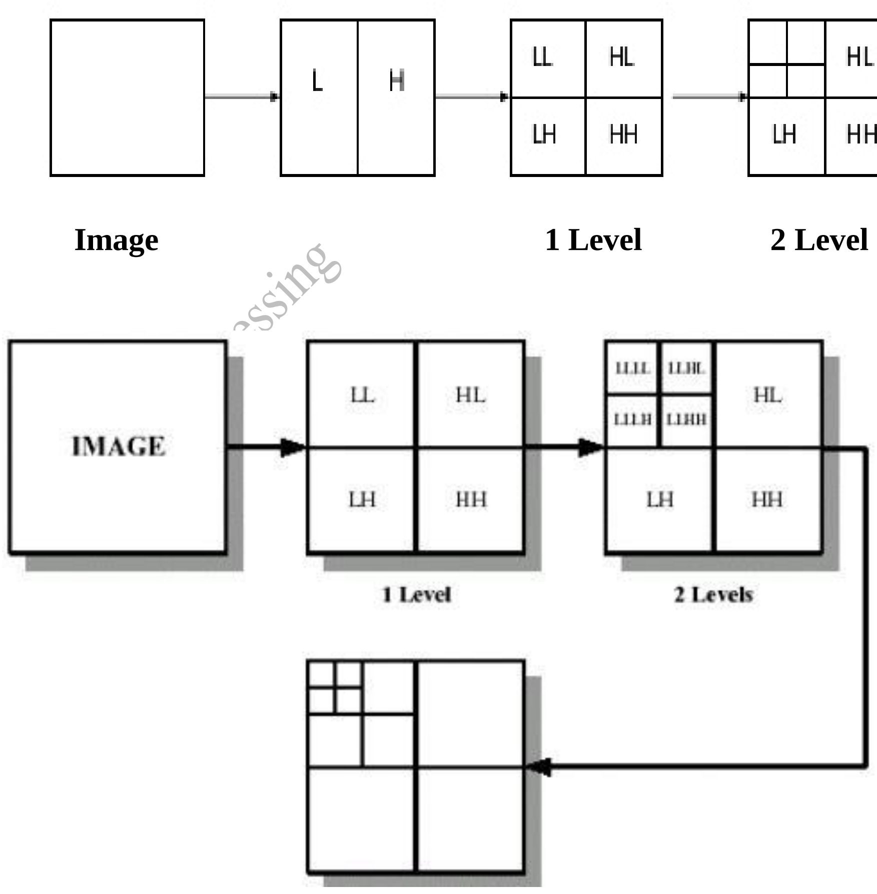
Image

The important feature of the DCT, the feature that makes it so useful in data compression, is that it takes correlated input data and concentrates its energy in just the first few transform coefficients.

3. Wavelet Transform

- Wavelet transforms are based on sub-sampling high and low pass filters.
- These filters are matched in such a way that they split the data into high and low pass bands with or without losing any information.
- Wavelet filters have been designed for a wide range of applications and many different sets.
- Wavelets are functions defined over a finite interval. The purpose of wavelet transform is to change the data from **Time-space domain** to **Time-frequency domain** which makes better compression results. The simplest form of wavelets, the Haar wavelet function is defined as:

. Nidaa F. Hassan



ssing

Figure (8.4): Block diagram of 3-level wavelet transform

The four bands that are formed are referred to as the Low-Low (LL), Low-High (LH), High-Low (HL) and High-High (HH). The LL band still has image-like information and so it is possible to apply the set of wavelet filters, in the same way as applied to the original image. This process of dividing the image into sub-bands can be continued as far as desired (to the resolution of the image), but for image compression it is usually only continued to 4 or 5 levels. A typical final image is shown in figure below:

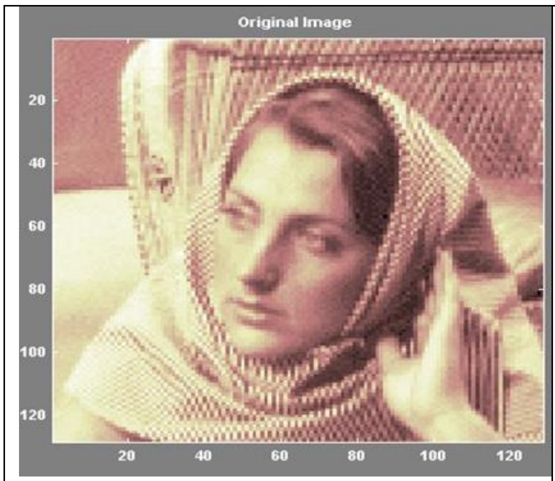
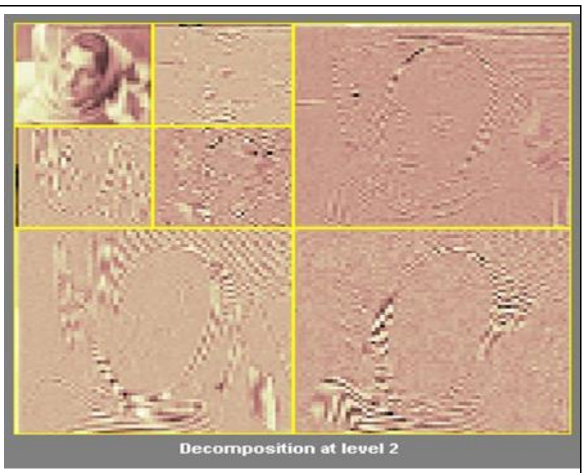
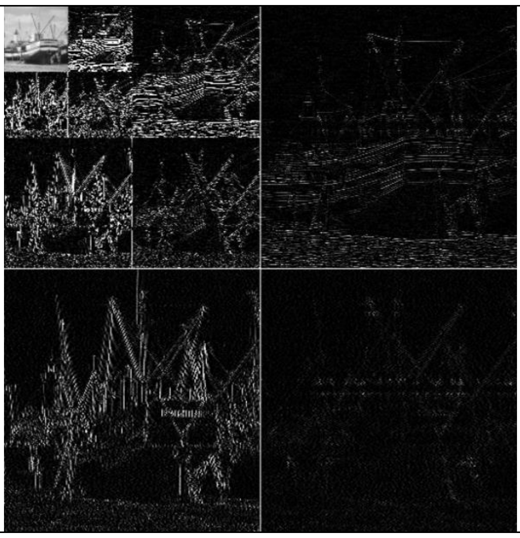
	
Orginal image	**WT in 2 levels**
	
Original image	**WT in 3 levels**

Figure (8.5): Example of Wavelet Transform

Haar Wavelet:

The Haar transform works mainly on pairs of data items from the signal begin processed and execute two step of calculation, the first step is:

$$\mathbf{L_i = 1/\sqrt{2}\ (X_{2i}+X_{2i+1})}$$

Which result an approximation (average) of the two data items, the second is:

$$\mathbf{H_i = 1/\sqrt{2}\ (X_{2i}-X_{2i+1})}$$

Which also results an approximation (difference) of the two data items

Where X is the signal data with length N

L is the Low sub –band with length N/2

H is the High sub-band with length N/2

i= 0…….N/2

The invers Haar transform formula is

$$\mathbf{X_{2i} = 1/\sqrt{2}\ (L_i+H_i)}$$

$$\mathbf{X_{2i+1} = 1/\sqrt{2}\ (L_i-H_i)}$$

Dr. Nidaa F.Hassan

<u>Example</u>: Haar Transform

4x4 image

137	**134**	**129**	**131**
135	**141**	**133**	**132**
138	**134**	**134**	**131**
135	**129**	**133**	**131**

Image Processing

L **H**

135	130	1	-1
138	132	-3	0
136	132	2	1
132	132	3	2

LL **HL** **LH** **HH**

136	131	-1	0
134	132	2	1
-1	-1	2	0
2	0	0	0

The <u>Haar wavelet</u> transform has a number of advantages:

- It is conceptually simple.
- It is fast.

- It is memory efficient, since it can be calculated in place without a temporary Array.
- It is exactly reversible without the edge effects that are a problem with other Wavelet transforms

Dr. Nidaa F.Hassan

Image Processing

Chapter Five
Image Fidelity Criteria

In lossy compression techniques, the decompressed image will not be identical to the original image. In such cases, we can define fidelity criteria that measure the difference between these two images. Fidelity criteria can be divided into two classes:

1) **Objective fidelity criteria**
2) **Subjective fidelity criteria**

The best instrument to measure image quality is the human eyes. Unfortunately, visual test are expensive to perform, so different statistical criteria can be used to measure the quality of the image.

1) The objective fidelity criteria:

The objective fidelity criteria are borrowed from digital signal processing and information theory and provide us with equations that can be used to measure the amount of distortion or error in the reconstructed (decompressed) image. Commonly used objective measures are:

- The mean square error (MSE)
- The signal to noise ratio (SNR)
- The peak signal to noise ratio (PSNR).

We can define the error between an original, uncompressed pixel value and the reconstructed (decompressed) pixel value as:

The mean square error is found by:

$$\mathbf{MSE} = \sqrt{1/N^2 \sum_{r=0}^{n-1} \sum_{c=0}^{n-1} [\ \bar{I}(r,c) - I(r,c)\ ]^2}$$

The signal to noise ratio is found by:

$$\mathbf{SNR} = \sqrt{\frac{\sum_{r=0}^{n-1} \sum_{c=0}^{n-1} [\ \bar{I}(r,c)\ ]^2}{\sum_{r=0}^{n-1} \sum_{c=0}^{n-1} [\ \bar{I}(r,c) - I(r,c)\ ]^2}}$$

Another related metric, the peak signal to noise ratio, is defined as:

$$\mathbf{PSNR} = 10 \operatorname{Log}_{10} \frac{(L-1)^2}{1/N^2 \sum_{r=0}^{n-1} \sum_{c=0}^{n-1} [\ \bar{I}(r,c) - I(r,c)\ ]^2}$$

Where L = the number of gray levels (e.g. for 8 bit, L=256)

The relationship between the SNR and MSE is ***reverse***. Its mean that when the MSE value of the image increase the SNR decrease and this will mean that the quality of the image is not good, but when the MSE value of the image decrease the SNR value will increase and in this case the quality of the image will be best.

These objective measures are often used in research because they are easy to generate and seemingly unbiased غير متحيز, but remember that these metrics are not necessarily correlated to our perception of an image.

2) The subjective fidelity criteria:

Subjective fidelity criteria require the definition of a qualitative scale to assess image quality. This scale can then be used by human test subjects to determine image fidelity.

Subjective testing is performed by creating a database of images to be tested, gathering a group of people that are representative of the desired population, and then having all the test subjects evaluate the images according to a predefined scoring criterion.

The results are then analyzed statistically, typically using the averages and standard deviations as metrics. Subjective fidelity measures can be classified into three categories:

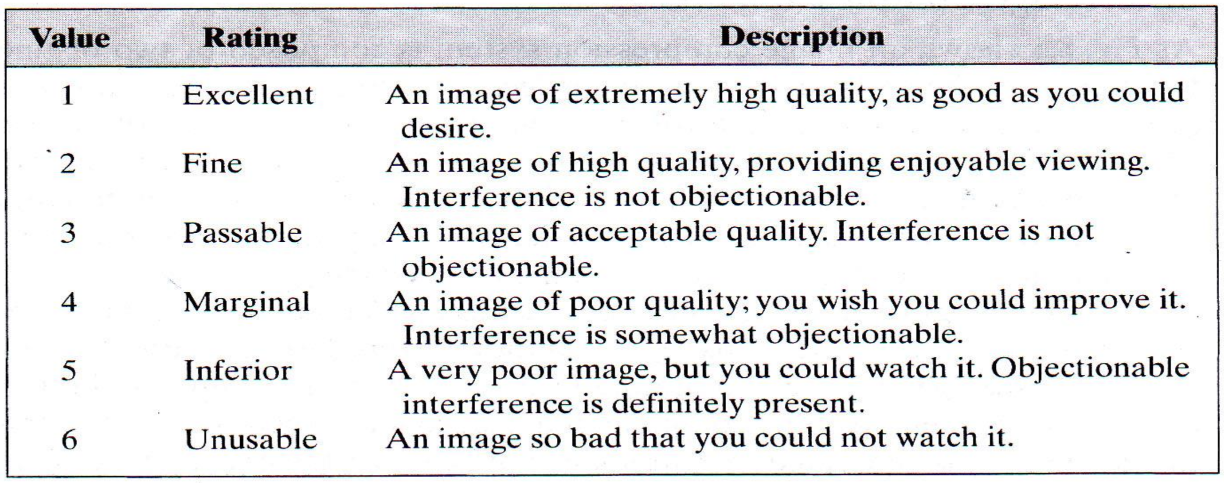
- The first type is referred to as impairment ضعف tests, where the test subjects score the images in terms of how bad they are.
- The second type is quality tests, where the test subjects rate the images in terms of how good they are.
- The third type is called comparison tests, where the images are evaluated on a side-by-side basis.

The comparison type tests are considered to provide the most useful results, as they provide a relative measure, which is the easiest metric for most people to determine.

Impairment and quality tests require an absolute measure, which is more difficult to determine in an unbiased fashion.

In Table (1) are examples of internationally accepted scoring scales for these three types of subjective fidelity measures.

Table (1) Subjective Fidelity Scoring Scales



Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.