University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Multimedia Fundamentals
اساسيات الوسائط المتعدد

Prof. Dr. Abdulamir Abdullah karim

أ. د. عبدالأمير عبدالله كريم

cs.uotechnology.edu.iq

Introduction to Multimedia

Multi media: means that computer information can be represented through audio, images, graphics, video and animation in addition to traditional media (text and graphics).

Video can be considered as an **integrated Multimedia** because it contains all the components of multimedia (images, sound and text).

Frame is any number of images in a time period (25-30 images per second) , those images are **similar (identical**) in characteristics.

1.1 What is Multimedia?

When different people mention the term **multimedia**, they often have quite different, or even opposing, viewpoints.

– ***A PC vendor***: a PC that has sound capability, video card, a DVD-ROM drive, and perhaps the superiority of multimedia-enabled microprocessors (e. g. Graphical Processing Unit **GPU**).

– ***A Computer Science (CS) student***: **applications** that use multiple modalities, including text, images, drawings (graphics), animation, video, sound including speech, and **interactivity**. From those applications, information hiding (**watermark**, **steganography**) which uses the available media as a cover to hide the secret data.

**** Multimedia and Computer Science:***

– Graphics, image processing, computer vision, data compression, networking - all have important contribution to make in multimedia at the present time.

1.2 Components of Multimedia

Multimedia involves multiple modalities of text, audio, images, drawings, animation, and video. Examples of how these modalities are **put to use**:

1. Video teleconferencing.
2. Distributed lectures for higher education (distance learning).
3. Tele-medicine.
4. Co-operative work environments – that allow business people to edit a shared document.
5. Searching in (very) large video and image databases for target visual objects (**Content Based Image or Video Retrieval**) .
6. Virtual Reality***:*** Interactive, computer-generated, three-dimensional graphics, delivered to the user through a head-mounted display.
7. Augmented Reality: placing real-appearing computer graphics and video objects into the real scenes.
8. Using **voice-recognition** to build an interactive environment application, say a **web browser**.

1.3 History of Multimedia

1. **Newspaper**: perhaps the ***first*** mass communication medium uses text, graphics, and images.
2. **Motion pictures**: conceived of in 1830's in order to observe motion too rapid for perception by the human eye.

3. **Wireless radio transmission**: Guglielmo Marconi, at Pontecchio, Italy, in 1895.
4. **Television**: the new medium for the 20th century, established video as a commonly available medium and has since changed the world of mass communications.
5. The **connection** between **computers** and ideas about **multimedia** covers what is actually only a short period:

1.4 Promising Multimedia Projects

Many exciting research projects are currently underway. Here are a few of them:

1. **Camera-based object tracking technology**: tracking of the control objects provides user control of the process.
2. **3D motion capture**: used for multiple actor capture so that multiple ***real actors*** in a ***virtual studio*** can be used to automatically **produce** ***realistic animated models*** with ***natural movement***.
3. **Multiple views**: from ***several cameras*** or from a single camera under ***differing lighting*** can accurately acquire data that gives both the **shape** and **surface** properties of materials, thus automatically generating synthetic **3D** ***graphics model***.
4. **3D capture technology**: allow **synthesis** of highly realistic **facial** animation from speech which is used in **fake videos**.
5. **Specific multimedia applications**: aimed at handicapped persons with **poor vision** capability and the **elderly** — a rich field of endeavor.
6. **Digital fashion**: aims to develop **smart clothing** that can communicate with other such enhanced clothing using wireless communication, so as to artificially enhance human interaction in a social setting.

7. **Electronic House call system**: an initiative for providing interactive ***health monitoring*** services to patients in their homes.
8. **Augmented Interaction applications**: used to develop **interfaces** between **real** and **virtual** humans for tasks such as augmented **storytelling (Hologram AVATAR)**.

1.5. Present Multimedia Applications

Examples of typical present multimedia applications include:

1. World Wide Web.

2. Multimedia courseware.

3. Distance learning

4. Video teleconferencing.

5. Virtual reality

6. Digital video editing and production systems.

7. Electronic newspapers/magazines.

8. On-line reference works: e.g. encyclopedias, games, etc.

9. Home shopping.

10. Interactive TV.

11. Interactive movies.

12. Video-on-demand.

In the below paragraphs, we describe some of the most important multimedia applications.

Video conferencing

Also cold teleconferencing, in which people in ***different geographical locations*** can have a meeting- can see and hear one another- using computers and communications. videoconferencing systems **rang** from ***videophones*** (is a telephone with high resolution screen and built-in camera that allows you to see the person you're calling) to group ***conference rooms*** with cameras and multimedia equipment to desktop systems with small ***video cameras, microphones, and speakers***.

Video conferencing may eliminate the need for some ***travel*** for the purpose of meeting and allow ***people who cannot travel*** to visit "in person". Many organizations use videoconferencing to take the place of face-to-face meetings.

Distance learning

Telecommunication technology is enabling many people to ***learn outside the class room***, a process called distance learning. Distance learning can be ***point-to-point*** (***synchronous)***, where students gathered at a specific location and the class is transmitted to them in ***real time (different place, same time).*** The students are able to see and hear the professor, and the professor can hear the students off-site and may be able to see them as well. The off-site locations may be around the same campus or across the world.

Distance learning may also be ***asynchronous*** (***different place, different time)***. Many courses are offered over the internet in prepackaged form.

Virtual reality : an emerging technology

There is no universal definition of virtual reality (VR). The most common ***definitions*** imply that virtual reality is interactive, computer-generated, three-dimensional graphics, delivered to the user through a head-mounted display.

Virtual reality, a computer-generated artificial reality, projects a person into a sensation of three-dimensional space. To put yourself into virtual reality, you need ***software*** and special headgear, and then you can add gloves, and later perhaps a special suit. The headgear; which is called head-mounted display- has two small video display screens, one for each eye, for creating the sense of three-dimensionality. Headphones pipe in stereophonic sound or even 3-D sound. Three-dimensional sound makes you think you are hearing sounds not only near each ear but also in various places all around you. The glove has sensors for collecting data about your hand movements. Once you are wearing these equipments, software gives you interactive sensory feelings similar to real-world experiences.

Benefits of Virtual Reality :

1. More than one person and even a large group can share and interact in the same environment.
2. VR thus can be a powerful medium for communication, entertainment, and learning.
3. The user can grasp and move virtual objects.
4. In virtual reality, a person "believes" that he or she is doing is real, even though it is artificially created.
5. The ***entertainment*** applications are obvious, but this capability can even be utilized for gaining a competitive ***business advantage***.

Sophisticated virtual reality systems are interactive and usually simulate real world phenomena. They often ***simulate sight, sound, and touch and combine these senses with computer-generated input to user's eyes, ears, and skin***. By using a head-mounted display, gloves, and bodysuit, or large projection images in simulator cabs, users can "enter" and interact with virtual or artificially generated environments.

With virtual reality, users can experience almost anything they want without ever leaving their chairs.

Virtual Reality Applications :

Military: training (pilots, drivers, shooting).

Medicine: training of surgeons (Surgeons can develop their skills through simulation on "digital patients).

Architecture: Simulate construction projects, create virtual objects on locations.

Entertainment Application.

Augmented Reality

Augmented Reality refers to Augmenting real world images with computer generated multimedia such as graphics, text, audio, and video.

One of the key technological challenges for creating an augmented reality is to maintain accurate **registration and tracking** between real word and computer generated object.

The main objective of AR is to **increase human sense of reality by adding some virtual information to the real scene**.

The Ultimate goal of any accurate registration method is: the computer generated virtual content (graphics, text, audio, and video), must be properly **overlaid on the real object**.

The three main **characteristics** of AR are:

1. Merge the virtual and real objects in the environment of real world.
2. Aligning (registering) and Tracking the virtual and real objects with each other in 3D environment.
3. Interactivity and in being a real time.

Virtual Reality Vs Augmented Reality

Augmented Reality	Virtual Reality
1. User maintains a sense of presence in real world. 2. System augments the real world scene with computer Generated multimedia. 3. Needs a mechanism to combine virtual and real worlds. 4. Hard to register real and virtual scene	1. The user is completely immersed in an artificial world and becomes isolated from the real environment. 2. Senses are under control of system. 3. Need a mechanism to feed virtual world to user. 4. Hard to make VR world interesting.

Introduction to Computer Graphics

1. **what's computer graphics**

Computer graphics is a set of all the means to deal with picture that can be seen by man and / or computer.

Man sees a picture when he becomes aware of a stimulus in a certain part of his brain (visual perception) , he then analysis it and interpret it.

A machine see a picture when it can analyze it and abstract from it features necessary for the tasks involved.

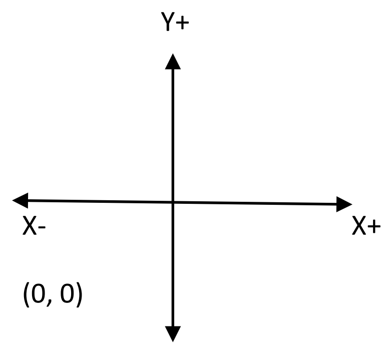
The term ***computer graphics*** involves using a computer to ***create*** and hold ***pictorial information*** and also to manipulate the display in different ways.

A computer is capable of sending its output to wide variety of devices, many of which are designed for special purposes. We will concern ourselves with devices that capable of producing graphical output e.g. display monitor , plotter, printer.

2. Cartesian coordinate system

A coordinate system provide a framework for ***translating geometric*** ideas into numerical expressions.

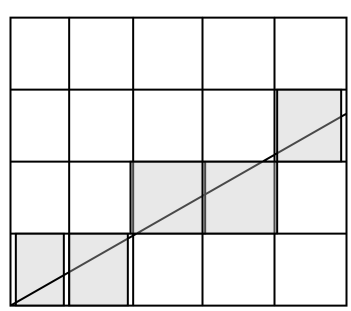
In a two-dimensional plane, we pick any point and single it out as a reference point called the ***origin***. Through the origin we construct two perpendicular number lines called axes. These are labeled the X axis and the Y axis. Any point in two dimensions in this X-Y plane can be specified by a pair of numbers, the first number is for the X axis, and the second number is for the Y axis.

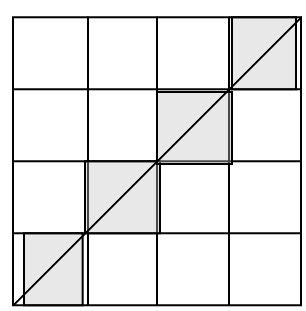


3. Device coordinate system

A Raster Display Monitor graphics devices can be considered a ***matrix of discrete cells*** each of which can be made bright. Thus it is a point plotting devices.

It is not possible except in special cases to directly draw a straight line from one addressable point, or pixel in the matrix to another addressable point, or pixel. The line can only approximated by a series of dots (pixels) close to the path of the line.





Only in the special cases of ***completely horizontal, vertical or 45*** degree lines will a straight line result. All other lines will appear as a series of ***stair steps***. This is called aliasing.

4. Drawing Lines

A fundamental task in graphics is drawing a line between two points. The graph unit ***provides a LINE statement that does this***. Still, you may find it useful to understand how such a routine works because , as you do more complicated graphics programming you will probably decide to write your own specialized routines.

4.1 Simple Line Drawing

The straight line algorithm is based on the algebraic equation for the straight line :

$$Y = a + b X$$

Where Y is the vertical coordinate, X is the horizontal coordinate, **a** is a constant factor, and **b** is the ***slope*** of the line. Once you have determined the value of **a** and **b** , drawing the line is easy.

To compute **a** and **b** ***you need two coordinate pairs***. In this code extract shown here, (x_1,y_1) and (x_2,y_2) are pair of coordinates representing points on the graphics screen.

```
dx := ( x2 – x1 );

dy := ( y2 – y1 );

if dx <> 0 then

   b := dy / dx;

else

   b:= 0;
```

$a := y_1 - x_1 * b;$

the **b** variable is defined as the difference between the **Y** coordinates divided by the difference between the **X** coordinates. If x_1 equal x_2 , then **b** is defined as zero. Once **b** is calculated , **a** is easily calculated using one of the two coordinate pairs.

With **a** and **b** both defined , the algorithm is complete. To draw the line, you need only trace along the x axis, computes the corresponding y coordinate , and plot the pixels.

The complete algorithm for line drawing is as below :

Algorithm 1 : a Simple Line Drawing

```
Procedure Switch( var  x,y : integer )
   begin
     t := x ;
     x := y ;
     y := t  ;
    end;

Begin
If  abs ( x1 – x2 ) > abs ( y1 – y2)  then
     begin
```

```
( * gaps between x's  is greater than y's .
     Trace horizontal  *)
If  x1 > x2  then
   begin
      Switch ( x1 , x2 );
      Switch ( y1 , y2 );
   end;

dx := ( x2 – x1 );
dy := ( y2 – y1 );
if dx <> 0 then
   b := dy / dx;
else
   b:= 0;
a:= y1 – x1 * b;

for x := x1 to x2 do
   begin
      y:= round ( a + x * b);
      PutPixel (x, y, color);
   end;
end
```

```
else
    begin
    ( * gaps between y's  is greater than x's .
       Trace vertically *)
    If  y1 > y2  then
    begin
      Switch ( y1 , y2 );
      Switch ( x1 , x2 );
    end;

    dx := ( x2 – x1 );
    dy := ( y2 – y1 );
    if dx <> 0 then
       b := dy / dx;
    else
       b:= 0;
    a:= y1 – x1 * b;

     for y := y1 to y2 do
     begin
        if  b <> 0 then
           x := round ( ( y – a ) / b);
```

```
            else
               x := 0;
               PutPixel (x, y, color);
         end;
      end;
   end;
```

Example 1:

Consider the line from (1,3) to (6,1) , use the simple line drawing to rasterize the line.

b = dy / dx = (1-3) / (6-1) = -2/5 = **- 0.4**

a = $y_1 - x_1$ * b = 3 – 1 × (– 0.4) = **3.4**

since | Δ x | > | Δ y | ⇨ gaps between x's is greater than y's

y:= round (a + x * b)

x1 =1   y1= 3.4 + 1 × (- 0.4) = 3

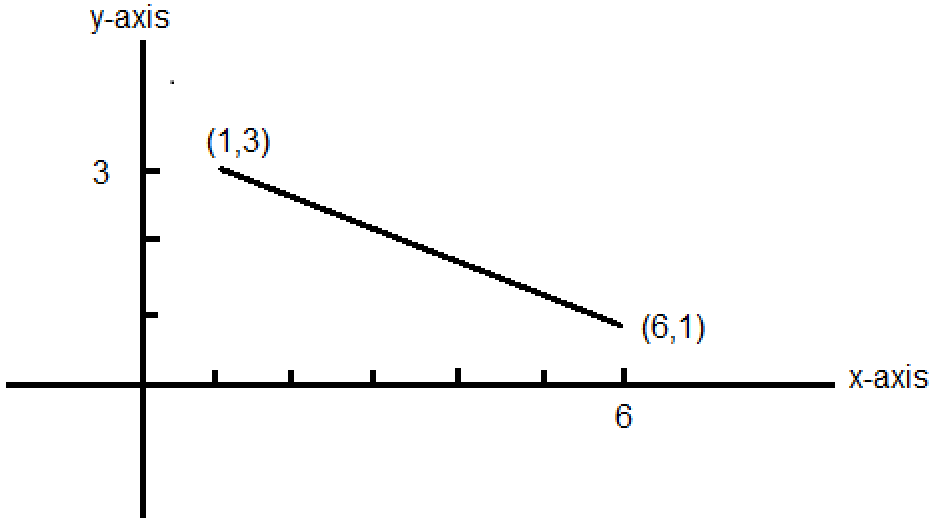
x2 =2   y2= 3.4 + 2 × (- 0.4) = 2.6= 3

x3 =3   y3= 3.4 + 3 × (- 0.4) = 2.2= 2

x4 =4   y4= 3.4 + 4 × (- 0.4) = 1.8= 2

x5 =5   y5= 3.4 + 5 × (- 0.4) = 1.4= 1

x6 =6   y6= 3.4 + 6 × (- 0.4) = 1



Example 2:

Consider the line from (1,1) to (4,6) , use the simple line drawing to rasterize the line.

b = dy / dx = (6-1) / (4-1) = 5/3 = **1.666**

a = $y_1 - x_1$ * b = 1 – 1 × (1.666) = **- 0.6**

since | Δ y | > | Δ x | ⟹ gaps between y's is greater than x's

x := round ((y – a) / b);

y1= 1 x1= 1- (- 0.6) / 1.666 = 0.96 = 1

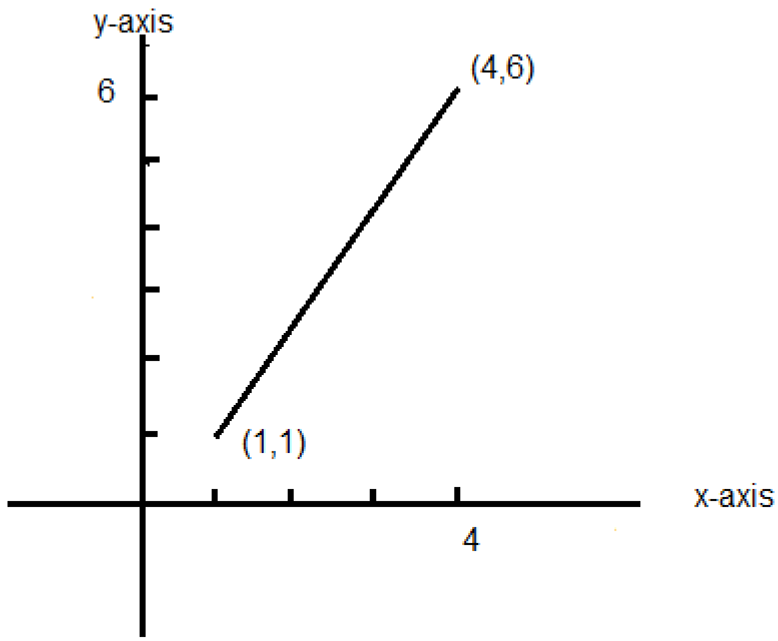
y2= 2 x2= 2- (- 0.6) / 1.666 = 1.56 = 2

y3= 3 x3= 3- (- 0.6) / 1.666 = 2.16 = 2

y4= 4 x4= 4- (- 0.6) / 1.666 = 2.7 = 3

y5= 5 x5= 5- (- 0.6) / 1.666 = 3.36 = 3

y6= 6 x6= 6- (- 0.6) / 1.666 = 3.96 = 4



4.2 DDA (Digital Differential Analyzer) algorithm

The DDA algorithm generates lines from their differential equations.

We calculate the ***length of the line in the X direction*** (number of pointes) by the equation:

$$ABS\ (X_2-X_1)$$

And calculate the ***length of the line in the Y direction*** (number of pointes) by the Equation :

$$ABS\ (Y_2-Y_1)$$

Where ABS is a function takes the positive of the arguments.

The Length estimates are equal to the ***larger*** of the magnitudes of the above two equations.

The increment steps (dX and dY) are used to increment the X and Y coordinates for the next pointes to be plotted

$$dX = \frac{X_2 - X_1}{\text{Larger Length}} \qquad dY = \frac{Y_2 - Y_1}{\text{Larger Length}}$$

Algorithm 2 : *DDA Line Drawing*

Start

If ABS(X_2*-*X_1*) > ABS (*Y_2*-*Y_1*) Then*

Length=ABS (X_2*-*X_1*)*

Else

Length=ABS (Y_2*-*Y_1*)*

$dX = (X_2 - X_1) / Length$

$dY = (Y_2 - Y_1) / Length$

$X = X_1 + 0.5 * Sign(\Delta X)$; $\Delta X = X_2 - X_1$

$Y = Y_1 + 0.5 * Sign(\Delta Y)$; $\Delta Y = Y_2 - Y_1$

For I=1 to Length

Begin

PutPixel *(Int(X) , Int (Y))*

X=X+dX

Y=Y+dY

End

Finish

Note :

1- Sign function returns : -1 if its argument is < 0
: 0 if its arguments is = 0
: +1 if its arguments is > 0

Ex. Sign(-10) = -1 ; Sign (5) = 1

Using the Sign function makes the algorithm work in all quadrants.

2- Int function works as follow :
   Int (8.5) = 8

   Int (-8.5) = -9

Example 1 : Consider the line from (0,0) to (5,5)

Use DDA to rasterizing the line.

Sol 1 :

X_1=0 ; Y_1=0 ; X_2=5 ; Y_2=5 ; Length=5

dX=1 ; dY=1 ; X=0.5 ; Y=0.5

I	Plot	X	Y
		0.5	0.5
1	(0,0)	1.5	1.5
2	(1,1)	2.5	2.5
3	(2,2)	3.5	3.5
4	(3,3)	4.5	4.5
5	(4,4)	5.5	5.5

Note : the Int function of X and Y are used in plotting the line.

This would normally have the effect of ***truncating*** rather than ***rounding*** so we initialize the DDA with the value 0.5 in each of the fractional parts to achieve true rounding. One advantage of this arrangement is that it allows us to detect changes in X and Y and hence to avoid plotting the same point twice

Example 2 : Consider the line from (0,0) to (-8,-4) ; evaluate the DDA algorithm

Sol 2 :

$X_1=0$; $Y_1=0$; $X_2=-8$; $Y_2=-4$; Length =8

dX=-1 ; dY=-0.5 ; X=-0.5 ; Y=-0.5

i	plot	X	Y
		-0.5	-0.5
1	(-1,-1)	-1.5	-1
2	(-2,-1)	-2.5	-1.5
3	(-3,-2)	-3.5	-2
4	(-4,-2)	-4.5	-2.5
5	(-5,-3)	-5.5	-3
6	(-6,-3)	-6.5	-3.5
7	(-7,-4)	-7.5	-4
8	(-8,-4)	-8.5	-4.5

5. Circle Drawing algorithms

In addition to drawing a straight line we need to draw a circle.

We can draw a circle using the following equations :

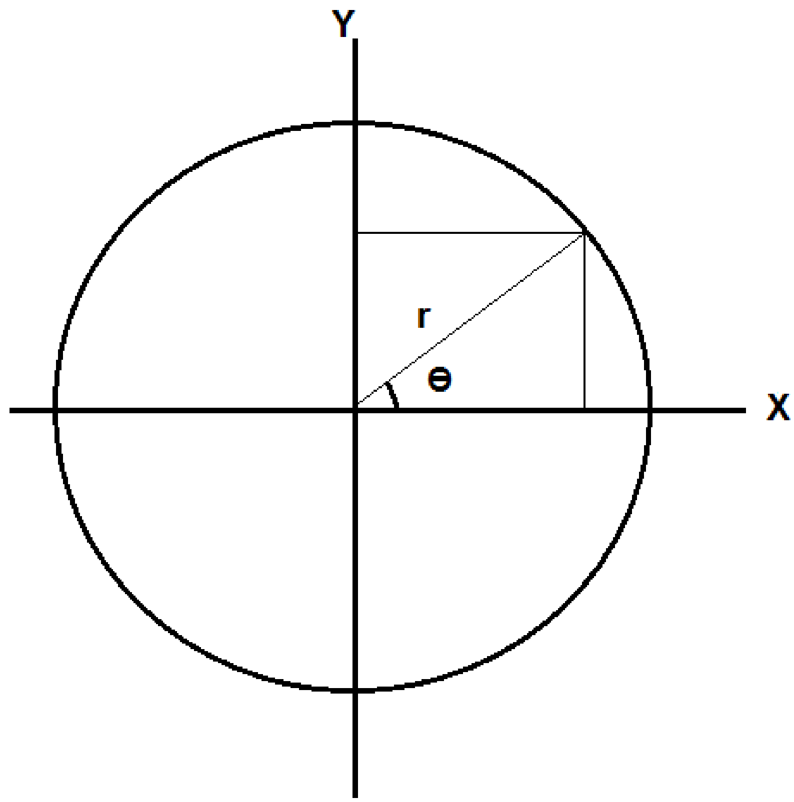
$x = r \cos(\Theta)$

$y = r \sin(\Theta)$

Where Θ vary from 0^{o} to 360^{o}.

Algorithm 3 : *a Simple Circle Drawing*

```
r = 300;
pia = 3.1428;
for  i = 0 to 360;
     theta =  i * pia / 180 ;    /* to convert from degree to radius */
     x =  r * cos ( theta );
     y =  r * sin  ( theta );
     putpixel( int(x) , int (y) );
 next ;
```



6. Two Dimension Transformation

Fundamental to all computer graphics systems is the ability to simulate the movement and the manipulation of objects in the plane. These processes are described in terms of:

1-Translation

2-Scaling

3-Rotation

4-Reflection

5-Shearing

Our object is to describe these operations in mathematical form suitable for computer processing.

6.1 Translation

Is a straight line movement of an object from one position to another.

If we have a point at (X,Y), that to be translated to a new position (X' , Y') , we must add a translation distance T_X and T_Y to the original point(X, Y) as follow :

$X'=X+T_X$

$Y'=Y+T_Y$

If T_X is positive then the point moves to right

If T_X is negative then the point moves to left

If T_Y is positive then the point moves up (in PC moves down)

If T_Y is negative then the point moves down (in PC moves up)

The transformation of Translation can be represented by (3*3) matrix:

$$[X' \quad Y' \quad 1] = [X \quad Y \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_X & T_Y & 1 \end{bmatrix}$$

Example1 : Move the line (-4,3) , (9,-6) 3 units in the X direction and 2 units in the Y direction

Solution:

$T_X=3$, $T_Y=2$

First point

$X'_1= -4 + 3= -1$

$Y'_1= 3 + 2= 5$

Second point

$X'_2= 9 + 3= 12$

$Y'_2= -6 + 2= - 4$

By using the matrix representation

X'_1	Y'_1	1
X'_2	Y'_2	1

=

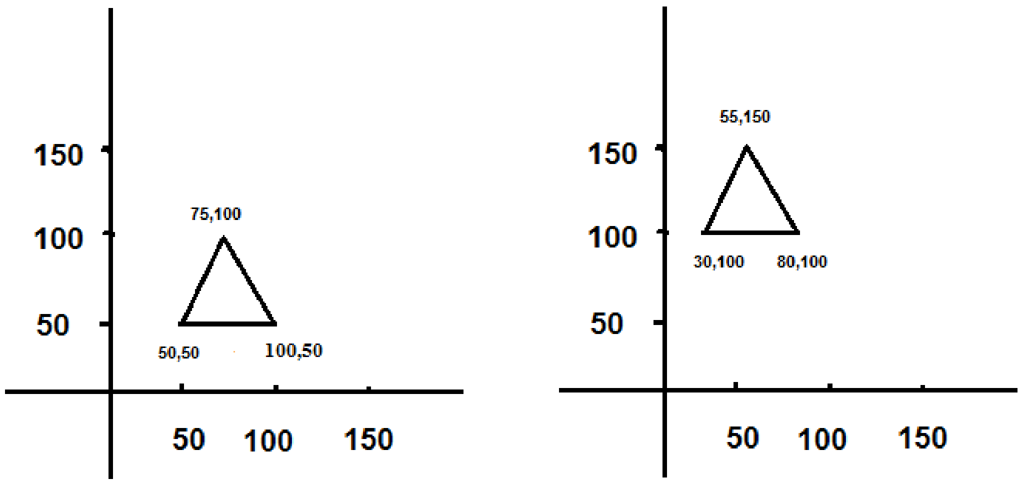
-4	3	1
9	-6	1

*

1	0	0
0	1	0
3	2	1

Example 2

Consider $T_X = -20$, $T_Y = 50$, apply translation to the figure below :



6.2 Scaling

A transformation to alter the size of an object is called scaling.

Scaling is the process of expanding or compressing the dimensions of an object (changing the size of an object). The size of an object can be change by multiplying the points of an object by scaling factor.

If SF (scale factor) > 1 then the object is enlarged

If SF (scale factor) < 1 (Not Negative) then the object is compressed

If SF (scale factor) = 1 then the object is unchanged

S_X is the scale factor in the X direction

S_Y is the scale factor in the Y direction

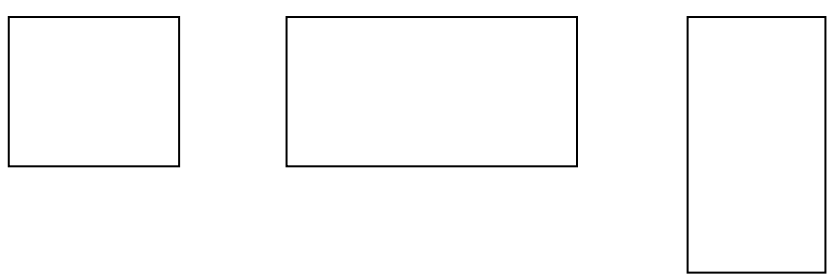
To scale a point P(X,Y) we use the equations:

$X' = X * S_X$

$Y' = Y * S_Y$

Where S_X and S_Y any real positive number.

If S_X and S_Y have the same value (Sx=Sy) a uniform scaling is produced.



Original shape $S_X = 2, S_Y = 1$

$S_X = 1 , S_Y = 2$

Using the matrix representation we can write the scaling equation as :

$$\begin{bmatrix} X' & Y' & 1 \end{bmatrix} = \begin{bmatrix} X & Y & 1 \end{bmatrix} * \begin{bmatrix} S_X & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

<u>Example 1</u>: Scale the rectangle (12,4),(20,4),(12,8),(20,8) with $S_X=2, S_Y=2$

Solution : (By using the equations)

For the point (12,4)

$X'_1= 12 * S_X =12 * 2 = 24$

$Y'_1= 4 * S_Y = 4 * 2 = 8$

For the point (20,4)

$X'_2= 20 * S_X =20 * 2 = 40$

$Y'_2= 4 * S_Y = 4 * 2 = 8$

For the point (12,8)

$X'_3= 12 * S_X =12 * 2 = 24$

$Y'_3= 8 * S_Y = 8 * 2 = 16$

For the point (20,8)

$X'_4= 20 * S_X = 20 * 2 = 40$

$Y'_4 = 8 * S_Y = 8 * 2 = 16$

Solution : (By using matrices)

X'_1	Y'_1	1
X'_2	Y'_2	1
X'_3	Y'_3	1
X'_4	Y'_4	1

=

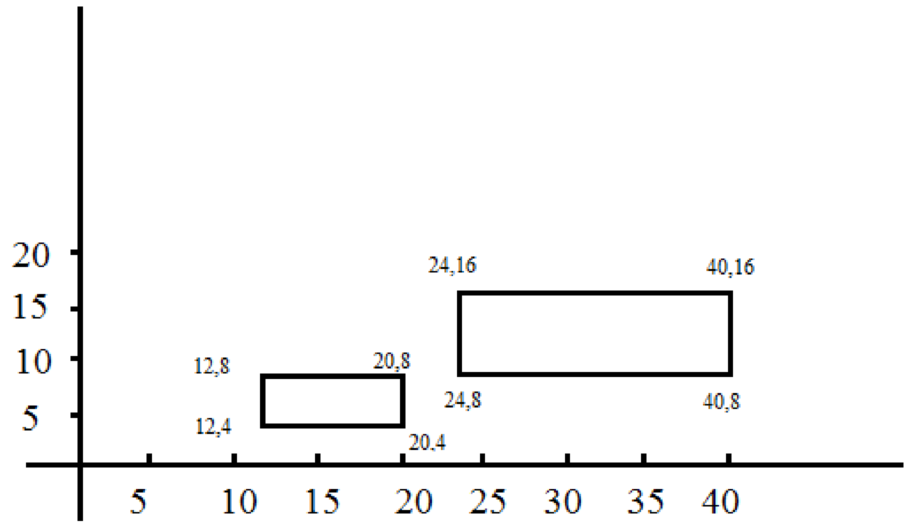
12	4	1
20	4	1
12	8	1
20	8	1

*

2	0	0
0	2	0
0	0	1

=

24	8	1
40	8	1
24	16	1
40	16	1



Notice that after a scaling transformation is performed, the new object is located at a different position relative to the origin. In fact, in scaling transformation the only point that remains fixed is the origin.

If we want to let one point of an object that remains at the same location (fixed), scaling can be performed by three steps:

1- Translate the fixed point to the origin, and all the points of the object must be moved the same distance and direction that the fixed point moves.

2- Scale the translated object from step one.

3- Back translate the scaled object to its original position

Example 2: Scale the rectangle (12,4),(20,4),(12,8),(20,8) with S_X =2, S_Y=2 so the point (12,4) being the fixed point.

Solution:

1- Translate the object with T_X= -12 and T_Y= -4 so the point (12,4) lies on the origin

(12,4) ===> (0,0)

(20,4) ===> (8,0)

(12,8) ===> (0,4)

(20,8) ===> (8,4)

2- Scale the object by S_X=2 and S_Y=2

(0,0) ===> (0 , 0)

(8,0) ===> (16 , 0)

(0,4) ===> (0 , 8)

(8,4) ===> (16 , 8)

3- Back translate the scaled object with T_X= 12 and T_Y= 4

(0 , 0) ===> (12 , 4)

(16 , 0) ===> (28 , 4)

(0 , 8) ===> (12 ,12)

(16 , 8) ===> (28 ,12)

For abbreviation, the three steps above can be analytically written as:

$X' = X_f + (X - X_f)\, S_X$;

$Y' = Y_f + (Y - Y_f)\, S_Y$;

3 1 2

١: نقل الشكل الى نقطة الأصل

٢ : تكبير الشكل

٣ : ارجاع الشكل الى مكانة الأصلي

Directly using the equation , we get :

$X' = X_f + (X - X_f)\, S_X$;

$X'_1 = 12 + (12 - 12)\, 2 = 12$

$X'_2 = 12 + (20 - 12)\, 2 = 28$

$X'_3 = 12 + (12 - 12)\, 2 = 12$

$X'_4 = 12 + (20 - 12)\, 2 = 28$

$Y' = Y_f + (Y - Y_f)\, S_Y$;

$Y'_1 = 4 + (4 - 4)\, 2 = 4$

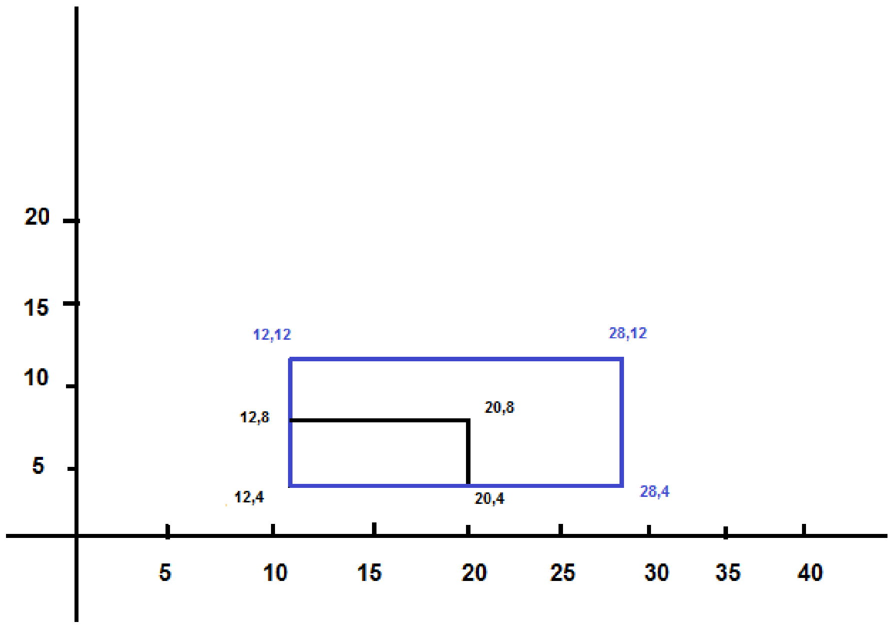
$Y'_2 = 4 + (4 - 4)\, 2 = 4$

$Y'_3 = 4 + (8 - 4)\, 2 = 12$

$Y'_4 = 4 + (8 - 4)\, 2 = 12$

So the new coordinates will be :

(12,4) , (28,4), (12,12), (28,12)



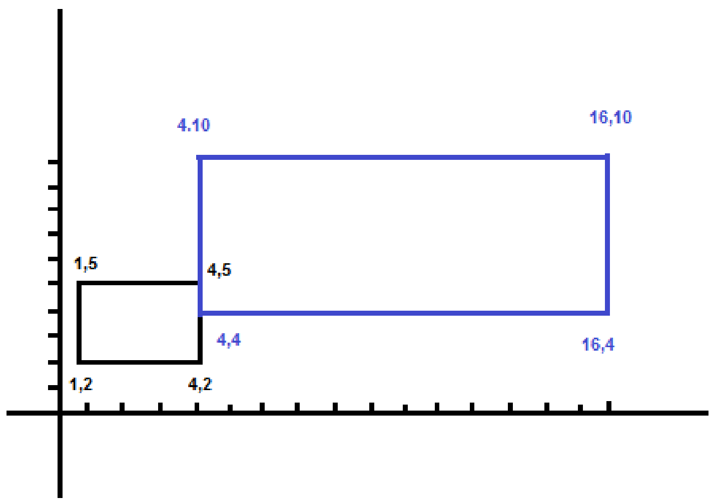
Example 3: Scale the square (1,2) , (4,2) , (1,5) , (4,5) with 4 units in the X-axis ,and 2 units in the Y-axis

Solution:

$$\begin{bmatrix} X'_1 & Y'_1 & 1 \\ X'_2 & Y'_2 & 1 \\ X'_3 & Y'_3 & 1 \\ X'_4 & Y'_4 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 2 & 1 \\ 1 & 5 & 1 \\ 4 & 5 & 1 \end{bmatrix} * \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$=$

4	4	1
16	4	1
4	10	1
16	10	1



Its obvious that the scale operation above cause a change in the rectangular position. So to avoid this drawback we have to apply the three steps of scaling , Directly using the last equations, we get :

$X' = X_f + (X - X_f)\, S_X$;

$X'_1 = 1 + (1 - 1)\, 4 = 1$

$X'_2 = 1 + (4 - 1)\, 4 = 13$

$X'_3 = 1 + (1 – 1) 4 = 1$

$X'_4 = 1 + (4 – 1) 4 = 13$

$Y' = Y_f + (Y - Y_f) S_Y;$

$Y'_1 = 2 + (2 – 2) 2 = 2$

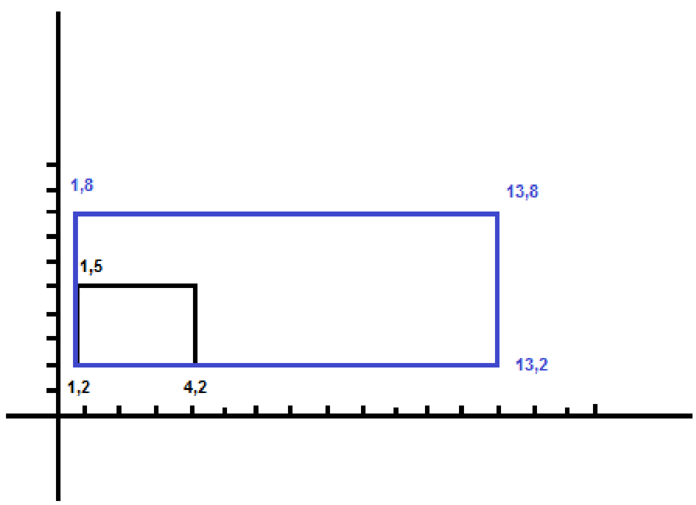
$Y'_2 = 2 + (2 – 2) 2 = 2$

$Y'_3 = 2 + (5 – 2) 2 = 8$

$Y'_4 = 2 + (5 – 2) 2 = 8$

So the new coordinates will be :

(1,2) , (13,2), (1,8), (13,8)



6.3 Reflection

If either the X or Y axis is treated as a mirror, the object has a mirror image or reflection. The reflected point Pnew is located the same distance from the mirror (the axis) as the original point P.

1:Reflection on the X axis

1	0	0
0	-1	0
0	0	1

OR Xnew= X

Ynew= -Y

2:Reflection on the Y axis

-1	0	0
0	1	0
0	0	1

OR Xnew= - X

Ynew= Y

3:Reflection on the origin

-1	0	0
0	-1	0
0	0	1

OR Xnew= - X

Ynew= -Y

4:Reflection on the line Y=X

0	1	0
1	0	0
0	0	1

OR Xnew= Y

Ynew= X

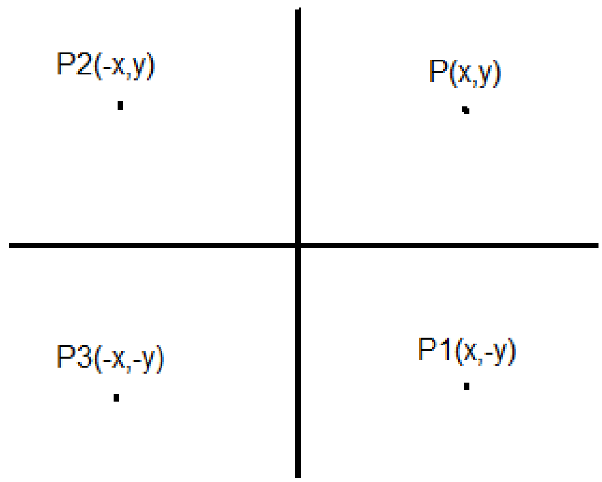
5:Reflection on the line Y=-X

0	-1	0

-1	0	0
0	0	1

OR Xnew= -Y

Ynew= - X



Example 1: Reflect the point P(3,2) in :: a- X axis; b- Y axis;

c-origin; d-line Y=X;

Solution : a-

3	2	1

*

1	0	0
0	-1	0
0	0	1

=

3	-2	1

b-

3	2	1

*

-1	0	0
0	1	0
0	0	1

=

-3	2	1

c-

3	2	1

*

-1	0	0
0	-1	0
0	0	1

=

-3	-2	1

d-

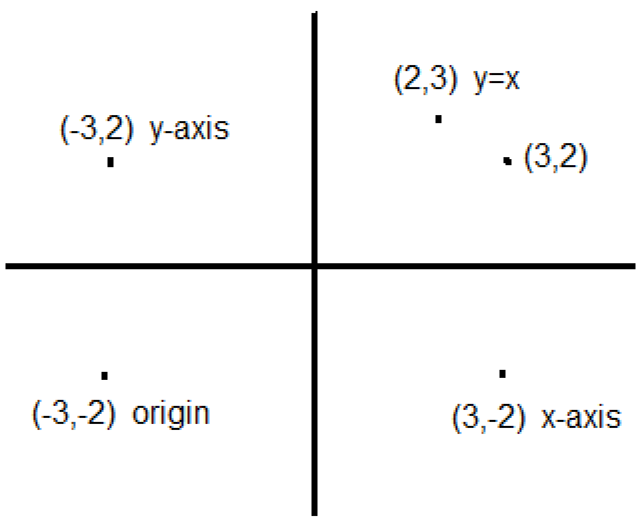
3	2	1

*

0	1	0
1	0	0
0	0	1

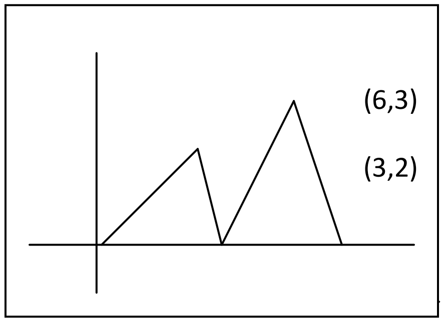
=

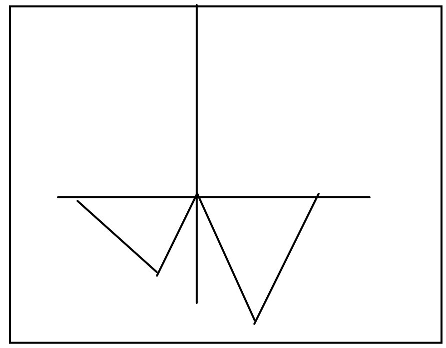
2	3	1



Example 2: What (3*3) matrix will change the center of the scene

to the origin, and reflect the mountains in the lake? [
assume the center of the scene is (4,0)]





Solution:

First: Translate by Tx= -4

1	0	0
0	1	0
-4	0	1

Second: Reflection on X axis

1	0	0
0	-1	0
0	0	1

Know multiply the two matrices :

1	0	0
0	1	0
-4	0	1

*

1	0	0
0	-1	0
0	0	1

The single matrix that perform Translation and Reflection is ➔

1	0	0
0	-1	0
-4	0	1

0	0	1
3	2	1
4	0	1
6	3	1
7	0	1

*

1	0	0
0	-1	0
-4	0	1

=

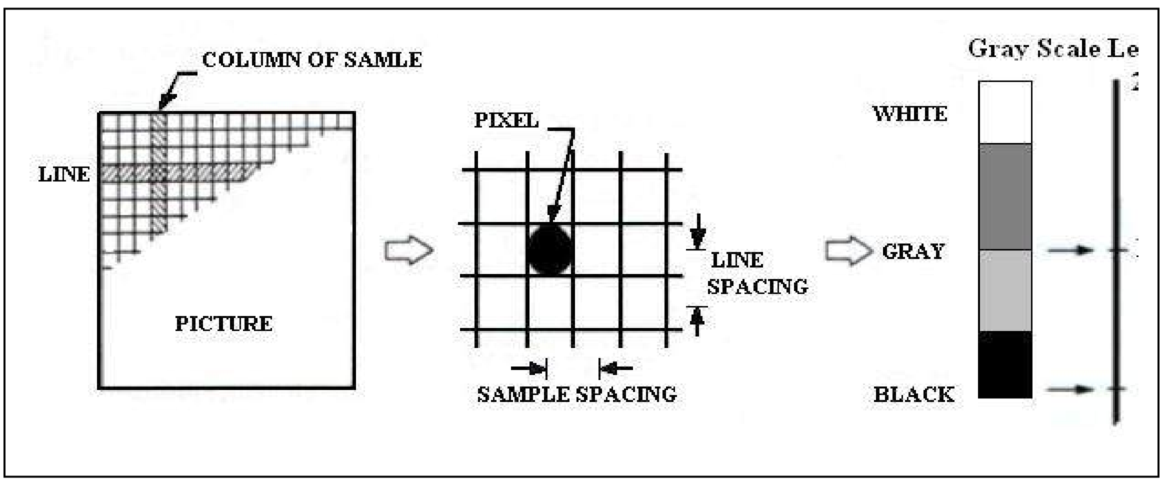
-4	0	1
-1	-2	1
0	0	1
2	-3	1
3	0	1

Introduction Digital Images

1. Digital Images

An image must be converted from it analogue form to numerical form before processing. This conversion process is called ***digitization*** , and a common form is illustrated in Figure(1).The image is divided into small regions called ***picture elements***, or ***pixel*** for short . The most common subdivision scheme is the rectangular ***sampling*** grid shown in Figure(1).The image is ***divided into horizontal lines made up of adjacent pixels*** . At each pixel location, the image (samples) brightness is ***quantized***. This step ***generates an integer at each pixel representing the brightness or darkness of the image at that point***. When this has been done for all pixel, ***the image is represented by a rectangular array of integers.*** Each pixel has a location or ***address*** (***Line or row number and sample or column number***) and an integer value called ***gray level***. This array of digital data is now a candidate for computer processing.

From above we can define ***Digital Image*** as a ***sampled, quantized function of two dimensions f(x,y)*** which has been generated by optical means, sampled in an equally spaced rectangular grid pattern , and quantized in equal intervals of gray levels.



Figure(1) Digitizing an Image

Thus a digital image is now a two-dimensional rectangular array of quantized sample value.

1.1 Sampling

The process of creating a digital image from date acquired by a camera or some other kind of imaging instrument, requires a ***two-dimensional*** pattern to represent the measurements (light intensity or color) that are made in the form of an image numerically.

The pattern can be described by a ***function f(x,y)***. For ***monochrome image***, the value of the function at any pair of coordinates, *x* and *y* is the intensity of the light detected at that point. In the case of ***color images*** *f(x,y)* is a vectored-value function

The function *f(x,y)* must be translated into a discrete array of numerical data if it is to undergo computer processing. Translation of *f(x,y)* into an appropriate numerical form is accomplished by the process of sampling and quantization

Sampling : is a process of measuring the value of the image function *f(x,y)* at discrete ***intervals in space***.

Or is the two dimensional pattern that is required to represent the image measurements (light intensity or color).

Each sample corresponds to a small square area of the image, known as a pixel. A digital image is a two-dimensional array of these pixels. Pixels are indexed by *x* and *y* coordinates, with *x* and *y* taking integer values.

Spatial Resolution

The spatial resolution of an image is the ***physical size*** of a pixel in that image; i.e., the ***real area*** in the scene that is represented by a single pixel in the image.

For a given field of view, ***dense sampling*** will produce a **high resolution** image in which there are many pixels, each of which represents the contribution of a very small part of the scene; ***coarse sampling***, on the other hand, will produce a **low resolution** image in which there are few pixels, each representing the contribution of a relatively large part of the scene to the image.

Spatial resolution dictates the amount of useful information that can be extracted from an image.

1.2. Quantization

If the image to be processed is analog (i.e., a voltage that changes with time) ***quantization*** is used to ***digitize it into integer numbers***.

It is usual to digitize the value of the image function $f(x,y)$ in addition to its spatial coordinates. The process of quantization ***involves*** replacing a continuously varying $f(x,y)$ (analog signals) with a ***discrete*** set of quantization levels (digital numbers) . The accuracy with which variations in $f(x,y)$ are represented is determined by the number of ***quantization levels*** that we use : ***the more levels we use, the better the approximation.***

Quantization: is the representation of the brightness of each pixel by an integer value. Since digital computer process number, it is necessary to reduce the continuous measurement value to discrete units and represent them by integer number.

Conventionally, a set of *n* quantization levels compromises the integers 0,1,2,……, *n-1*. ***0 and n-1*** are usually displayed or printed as ***black and white***. Respectively, with ***intermediate levels*** rendered in various ***shades of gray***. Quantization levels are therefore commonly referred to as ***gray levels***. The collective term for all the gray levels ranging from black to white, is a gray scale.

2. Image Representation

As we know, the human visual system receives an input image as a collection of spatially distributed ***light energy***; this form is called an ***optical image***. Optical images are the types we deal with everyday - cameras capture them, monitors display them, and we see them. We know that these ***optical images*** are represented as video information i***n the form of analog electrical signals and have seen how these are sampled to generate the digital image*** *f(x,y)*.

The digital image f(x,y) is represented as a two-dimensional array of data, where each pixel value corresponds to the brightness of the image at the point (r, c). In linear algebra terms, a two-dimensional array like our image model *f(x,y)* is referred to as a matrix, and one row (or column) is called a vector. This image model is for ***monochrome*** (one-color, this is what we normally refer to as black and white) image data, we also have other types of image data that require extensions or modifications to this model. Typically, these are multiband images (***color, multispectral***), and they can be modeled by a different *f(x,y)* function corresponding to each separate band of brightness information. The image types we will consider are : 1) binary, 2) gray-scale, 3) color, and 4) multispectral.

2.1 Binary Images

Binary images are the simplest type of images and can take on two values, typically black and white, or '0' and '1' A binary image is referred to as a ***1 bit/pixel*** image because it takes only 1 binary digit to represent each pixel.

Binary image applications

These types of images are most frequently used in computer vision applications where the only information required for the task is **general shape or outline**, information. For example:

1. To position a ***robotic*** gripper to grasp an object, to check a manufactured object for deformations, since robot has to work in **real time** , and working with Binary image decrease processing time.
2. In optical character recognition (OCR).
3. In text images.

Advantage of binary images

1. Reduce storage space (memory).
2. Decrease processing time.
3. Required less transmission time.

Binary images are often created from gray-scale images via a ***threshold*** operation where every pixel above the threshold value is turned white ('1'), and those below it are turned black ('0').

2.2 Gray-Scale Images

Gray-scale images are referred to as ***monochrome***, or one-color, images. They contain ***brightness information*** only, no color information. The number of bits used for each pixel determines the number of different brightness levels Available. The typical image contains ***8 bits/pixel*** data, which allows us to have ***256 (0-255) different brightness*** (gray) levels.

This representation provides more than adequate brightness resolution, in terms of the human visual system's requirements and provides a "noise margin" by allowing for approximately twice as many gray levels as required. Additionally, the ***8 bit representation*** is typical due to the fact that the byte which corresponds to 8 bit of data is the ***standard small unit*** in the world of digital computers.

2.3 Color Images

The use of color is important in image processing because:

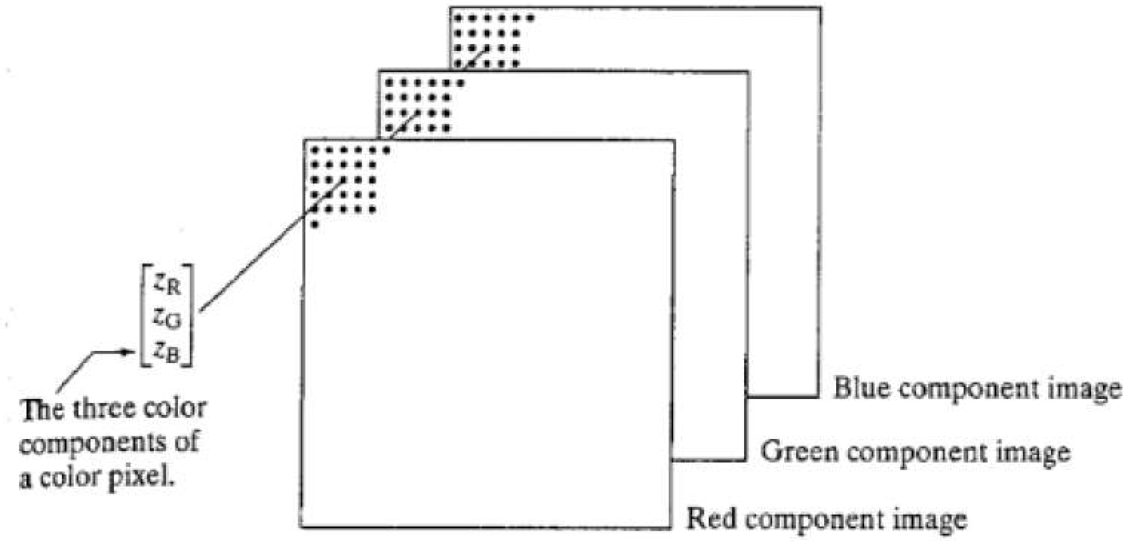
1. Color is a ***powerful descriptor*** that simplifies object identification and extraction.

2. Humans can ***discern thousands of color*** shades and intensities, compared to about only two dozen shades of gray.

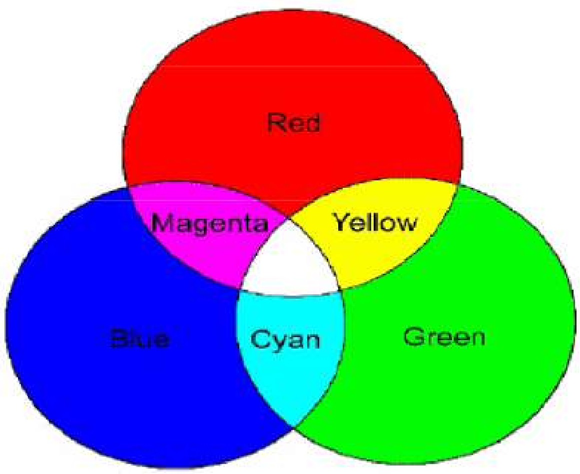
RGB Color Model

Color images can be modeled as three-band monochrome image data, where each band of data corresponds to a different color. The actual information stored in the digital image data is the brightness information in each spectral band. When the image is displayed, the corresponding brightness information is displayed on the screen by picture elements that emit light energy corresponding to that particular color ***which is generated from the mixture of those three bands.***

Typical color images are represented as Red, Green, and Blue, or RGB images. Using the 8-bit monochrome standard as a model, the corresponding color image would have 24 bits/pixel 8-bits for each of the three color bands (Red, Green, and Blue). In Figure (2) we see a representation of a typical RGB color image. Figure (2) illustrates that, in addition to referring to a row or column as a vector, we can refer to a single pixel's Red, Green, and Blue values as a color pixel vector (R,G,B).



(a)



(b)

Figure 2: (a) Scheme of RGB color image (b) Primary and secondary colors.

2.4 Multispectral images

Visible light is part of the ***electromagnetic*** spectrum: radiation in which the ***energy*** takes the form of waves of ***varying wavelength***. The values for the wavelengths of blue, green and red were set in 1931 by the CIE

(Commission International d'Eclairage), an organization responsible for color standards. Figure 4 illustrates this.

Multispectral images typically contain information ***outside*** the normal human perceptual range. This may **include** infrared, ultraviolet, X-ray, radar data. These are not images in the usual sense because the information represented is ***not directly visible*** by the human visual system. ***However, the information is often represented in visual form by mapping the different spectral bands to RGB components.*** If more than three bands of information are in the multispectral image, the dimensionality is reduced by applying a principal component's transform.

Sources for these types of images include ***satellite systems, underwater sonar system, various types of airborne radar, infrared imaging systems and medical diagnostic imaging systems***. The number of bands into which the data are divided is strictly a function of the sensitivity of the imaging sensors used to capture the images. For example, even the visible spectrum can be divided into many more than three bands; three are used because this mimics our visual system. ***Most of the satellites currently in orbit collect image information in two to seven spectral bands typically one to three are in the visible spectrum***, ***one or more are in the infrared region***, and ***some have sensors that operate in the radar range*** (see Figure 4). The newest satellites have sensors that collect image information in 30 or more bands. As the amount of data that needs to be transmitted, stored, and processed increases, the importance of topics such as compression becomes more and more apparent.

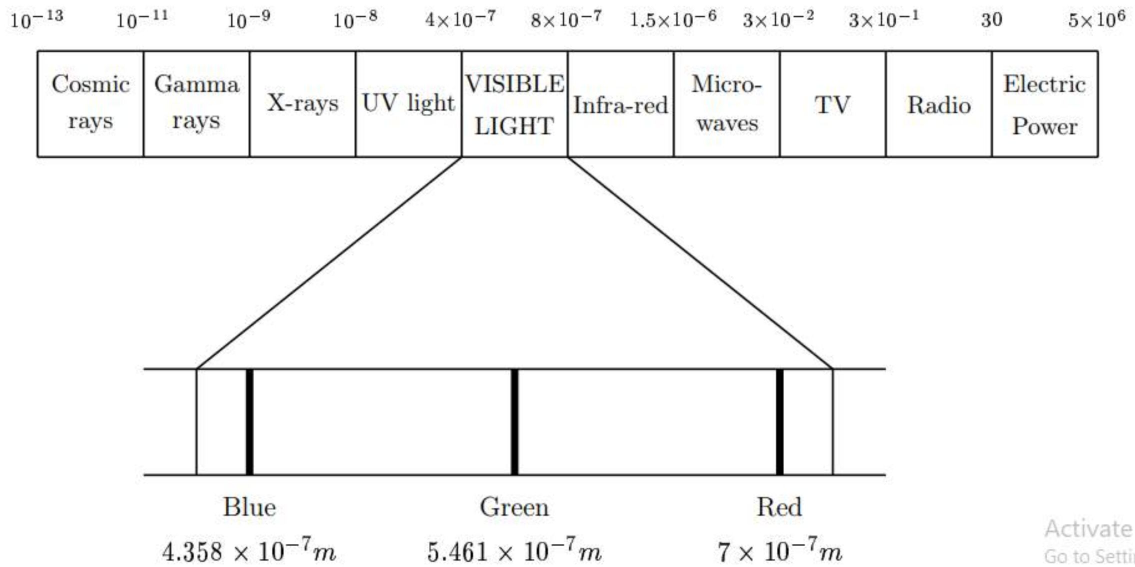
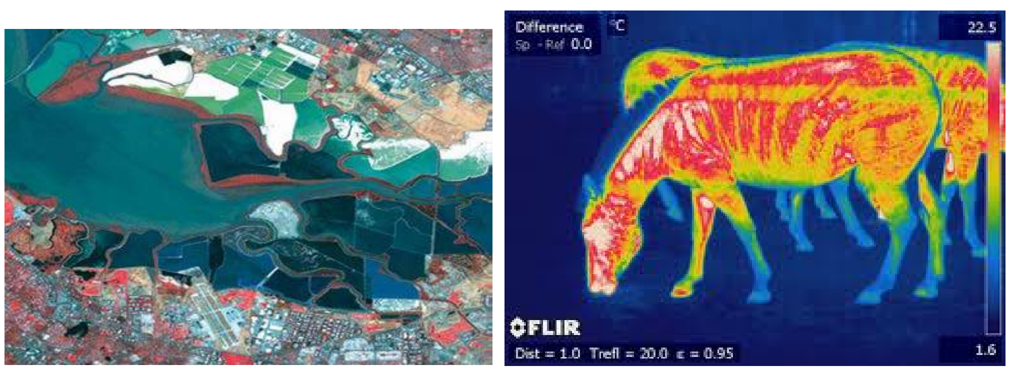


Figure (4) : The spectrum of electromagnetic radiation



A Multi spectral Image

3. Arithmetic and Logical Operations on Images (Image Algebra)

These operations are applied on ***pixel-by-pixel*** basis. So, to add two images together, we add the value at pixel (0 , 0) in image 1 to the value at pixel (0 , 0) in image 2 and ***store the result in a new image*** at pixel (0 , 0). Then we move to the next pixel and repeat the process, continuing until all pixels have been visited.

Clearly, this can work properly only if the two images have ***identical dimensions*.** If they do not, then combination is still possible, but a ***meaningful result*** can be obtained only in ***the area of overlap***. If our images have dimensions of w_1*h_1, and w_2*h_2 and we assume that their origins are aligned, then the new image will have dimensions w*h, where:

$w = \min(w_1, w_2)$

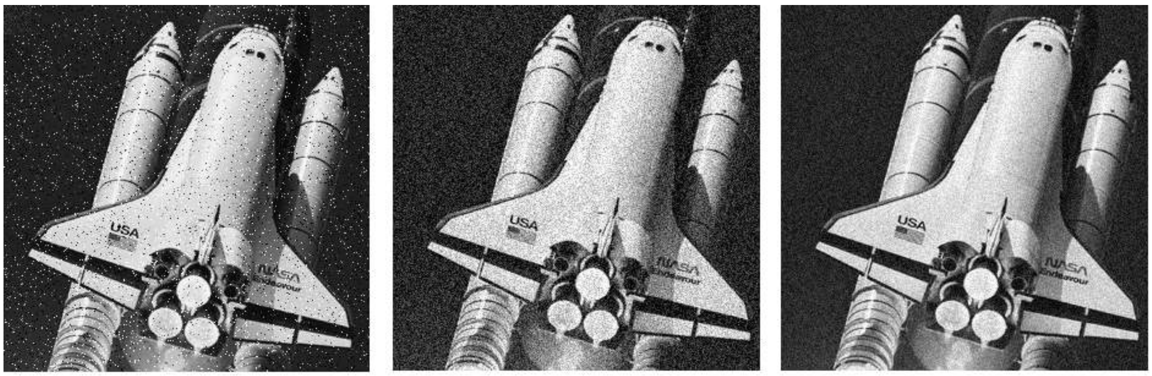
$h = \min(h_1, h_2)$

In the case of arithmetic operations, we must also ensure that the ***representation used*** for the output image is ***appropriate for the operation*** being performed. For example, ***the values produced when we add two 8-bit grey scale image, cannot , in general, be contained in an 8-bit range.***

3.1 Addition and Averaging

If we add two 8-bit grey scale images, then pixels in the resulting image can have values in ***the range* 0-510**. We should therefore choose a **16-bit *representation*** for the output image **or** ***divide every pixel's value by two***. If we do the later, then we are computing an average of the two images.

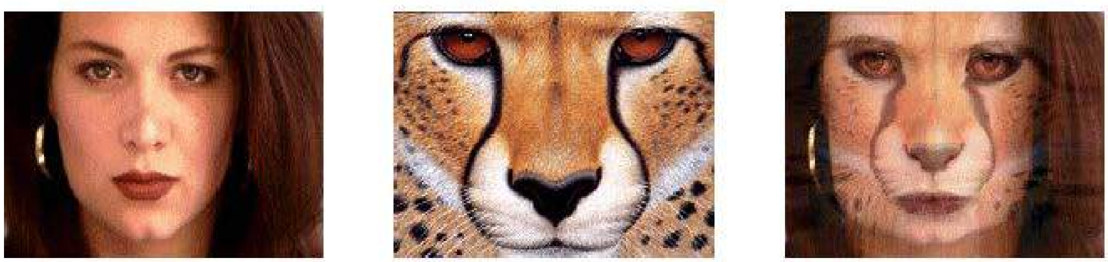
The main application of image averaging is ***noise removal***. Every image acquired by a real sensor is afflicted to some degree by ***random noise***. However, the level of noise represent in the image can be reduced, provided that the scene is static and unchanging, by the averaging of multiple observations of that scene. This works because the noisy distribution can be regarded as approximately symmetrical with a mean of zero. As a result, positive perturbations of a pixel's value by a given amount are just as likely as negative perturbations by the same amount, and there will be a tendency for the perturbations to cancel out when several noisy values are added.



(a) (b) (c)

Figure (5) a) noisy image b) average of five observation c) average of ten observation

Addition can also be used to ***combine the information of two images***, such as an ***image morphing and motion pictures***.



morphing: blending between two photographs

Algorithm 1: image addition

read input-image1 into in-array1;

read input-image2 into in- array2;

for i = 1 to no-of-rows do

for j=1 to no-of-columns do

begin

```
        out-array (i,j) = in-array1(i,j) + in-array2(i,j);
        if ( out-array (i,j) > 255 )  then   out-array (i,j) = 255;
    end
write out-array to out-image;
```

3.2 Subtraction

Subtracting two 8-bit grayscale images can produce values between **-225 and +225.** This necessitates the use of ***16-bit signed integers*** in the output image – unless ***sign is unimportant***, in which case we can simply take the ***modulus of the result*** and store it using 8-bit integers:

$$g(x,y) = |f_1 (x,y) - f_2 (x,y)|$$

The main application for image subtraction is in ***change detection*** (or ***motion detection***). If we make two observations of a scene and compute their difference using the above equation, then changes will be indicated by pixels in the difference image which have ***non-zero values***. Sensor noise, slight changes in illumination and various other factors can result in small differences which are of no significance so it is usual to apply a threshold to the difference image. Differences below this threshold are set to zero. Difference above the threshold can, if desired, be set to the maximum pixel value. Subtraction can also be used in ***medical imaging to remove static background information***.

Algorithm2: image subtraction

```
read input-image1 into in-array1;
read input-image2 into in- array2;
for  i = 1   to no-of-rows do
```

```
for j=1 to no-of-columns do
   begin
      out-array (i,j) = in-array1(i,j) - in-array2(i,j);
      if ( out-array (i,j) < 0 )  then   out-array (i,j) = 0;
  end
```

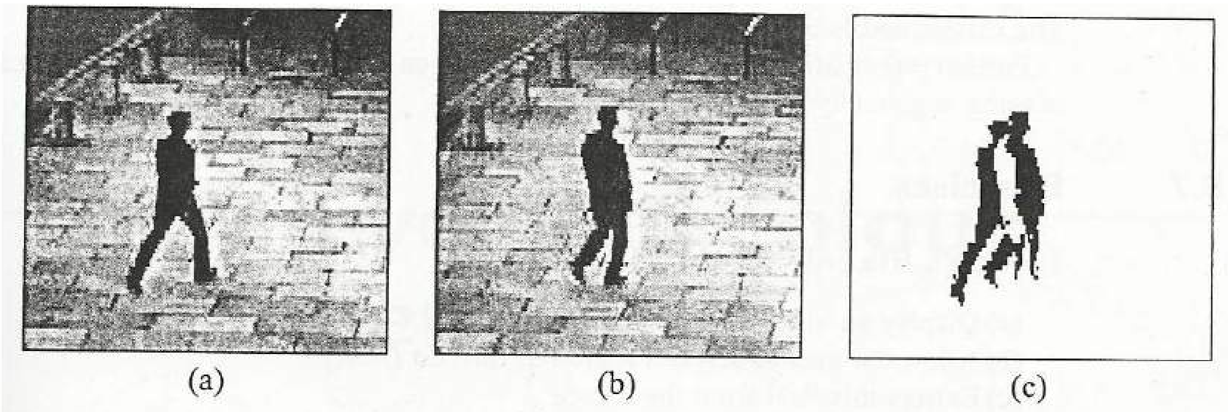


Figure (6)   a, b) two frames of video sequence   c) their difference

3.3 Multiplication and Division

Multiplication and division can be used to ***adjust brightness*** of an image. **Multiplication** of pixel values by a ***number greater than one will brighten the image***, and **division** by a ***factor greater than one will darken the image***. Brightness adjustment is often used as a ***preprocessing step in image enhancement.***

One of the principle uses of image multiplication (or division) is to ***correct grey-level shading*** resulting from non uniformities in illumination or in the sensor used to acquire the image.

(a) **(b)** **(c)**

Figure (7) a) original image b) image multiplied by 2 c) image divided by 2

3.4 Logical Operation:

Logical operations apply ***only to binary images***, whereas arithmetic operations apply to ***multi-valued pixels***. Logical operations are basic tools in binary image processing, where they are used for tasks such as ***masking***, ***feature detection***, ***shape analysis*** and ***obtain the similarity or difference between two templates***. Logical operations on entire image are performed **pixel – by – pixel**. Because the AND operation of two binary variables is 1 only when both variables are 1, the result at any location in a resulting AND image is 1 only if the corresponding pixels in the two input images are 1. As logical operation involve only one pixel location at a time, they can be done in place, as in the case of arithmetic operations. The XOR (exclusive OR) operation yields a 1 when one or other pixel (but not both) is 1, and it yields a 0 otherwise. The operation is unlike the OR operation, which is 1, when one or the other pixel is 1, or both pixels are 1.

AND

Input	1	0	1	0
Mask	1	1	0	0
output	1	0	0	0

OR

Input	1	0	1	0
Mask	1	1	0	0
output	1	1	1	0

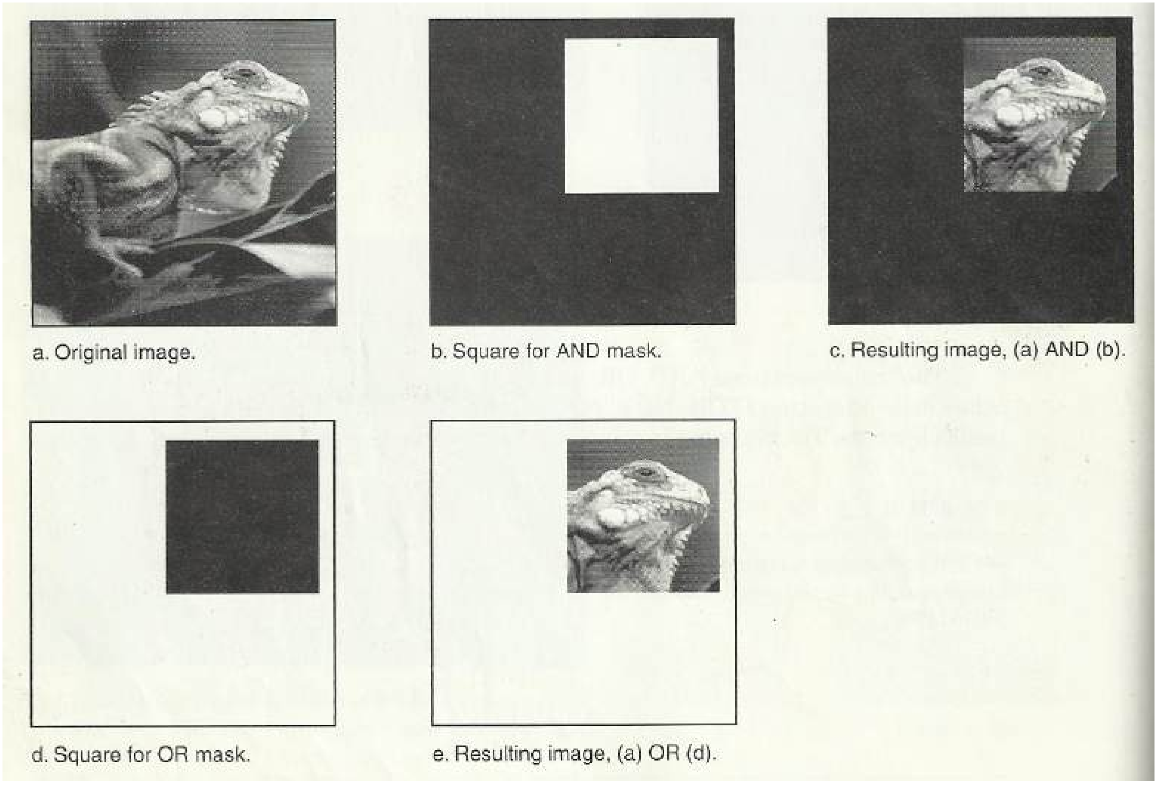
XOR

Input	1	0	1	0
Mask	1	1	0	0
output	0	1	1	0

Logical **AND & OR** operations are useful for the ***masking and cropping*** of images. For example, if we compute the **AND** of a binary image with some other image, then pixels for which the corresponding value in the binary image is 1 will be ***preserved***, but pixels for which the corresponding binary value is 0 will be set to 0 ***(erased)*** . Thus the binary image acts as a "***mask***" that ***removes*** information from certain parts of the image.

On the other hand, if we compute the **OR** of a binary image with some other image , the pixels for which the corresponding value in the binary image is 0 will be ***preserved***, but pixels for which the corresponding binary value is 1, will be set to 1 (***cleared***).

So, masking is a simple method to extract a region of interest from an image.



Figure(8) : image masking

In addition to masking, logical operation can be used in feature detection. Logical operation can be used to ***compare between two images***, as shown below:

AND ^

This operation can be used to find the ***similarity between white regions*** of two different images (it required two images).

g (x,y) = a (x,y) ^ b (x,y)

Exclusive OR ⊗

This operator can be used to find the ***differences between white regions*** of two different images (it requires two images).

g (x,y) = a (x,y) ⊗b (x,y)

NOT

NOT operation can be performed on ***grey-level*** images, it's applied on only one image, and the result of this operation is the ***negative*** of the original image.

$$g(x,y) = 255 - f(x,y)$$

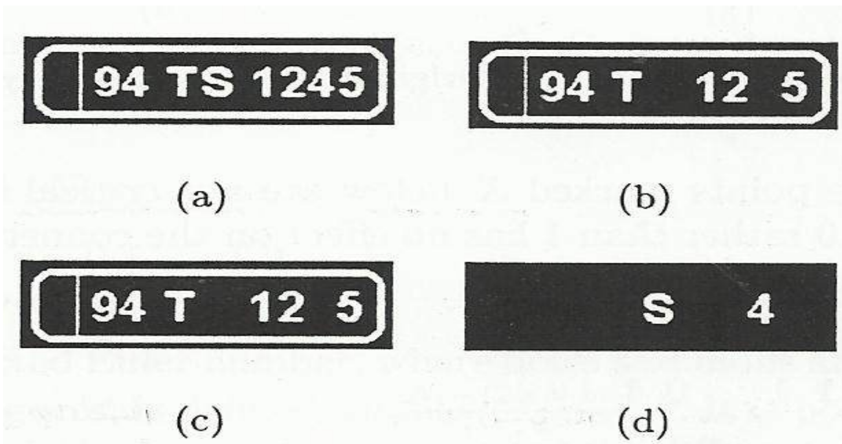


Figure (9) a) input image a(x,y) b) input image b(x,y)

c) a(x,y) ^ b(x,y) d) a(x,y) ⊗b(x,y)

4. Image Histogram

The gray level ***histogram*** is a function showing, for each gray level, the ***number of pixels*** in the image that have ***that gray level***. The abscissa is gray level and ordinate is the frequency of occurrence (number of pixels). This function summarizes the gray level counted of an image. While the histogram of any image contains considerable information. ***Certain types of images are completely specified by their histograms***.

The histogram of an image records the frequency distribution of gray levels in the image. The histogram of an ***8-bit image*** can be thought of as a ***table with 256 entries, or "bins", indexed from 0 to 255***. In bin 0 we record the number of times a gray level of 0 occurs; in bin 1 we record the number of times a gray level of 1 occurs, and so on , up to bin 255.

Algorithm 3 shows how we can accumulate in a histogram from an image. Figure 10 shows an image and its histogram computed using this algorithm.

ALGORITHM 3: Calculating of an image Histogram

Create an array histogram with 2^8 elements.

For all gray levels, I, do

    Histogram [I] =0

End for

For all pixels coordinates, x and y, do

    Increment histogram [f (x,y)] by 1

End for

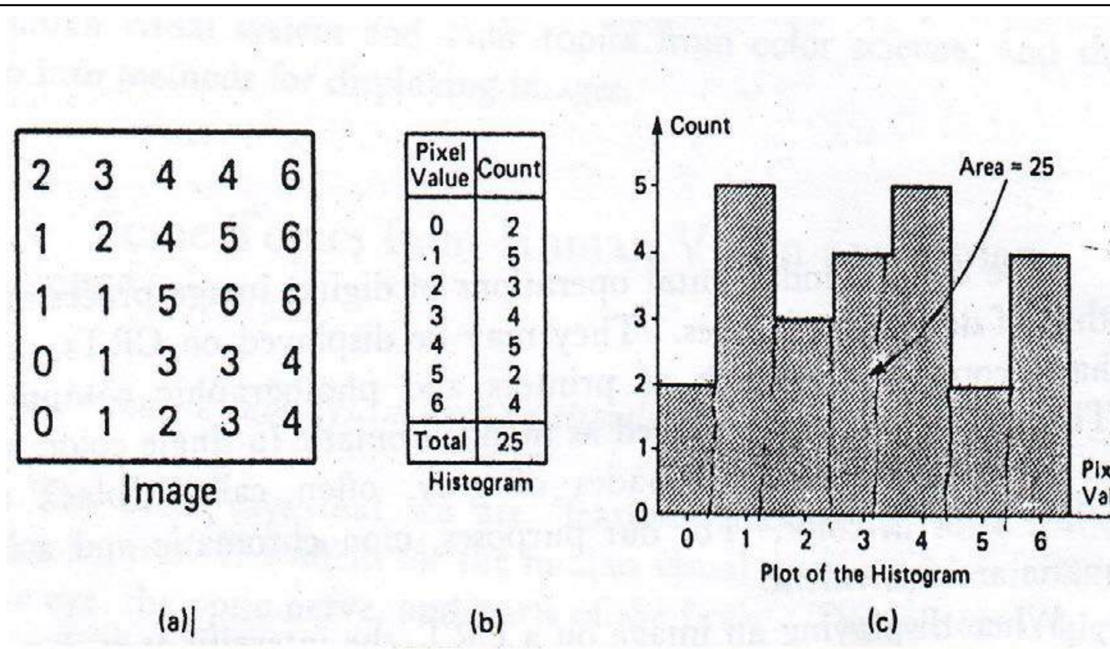
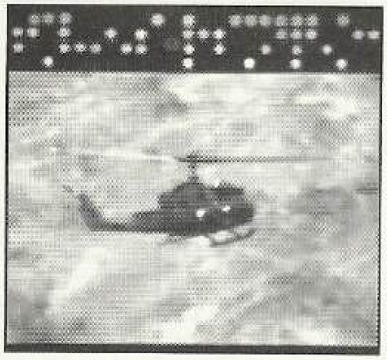


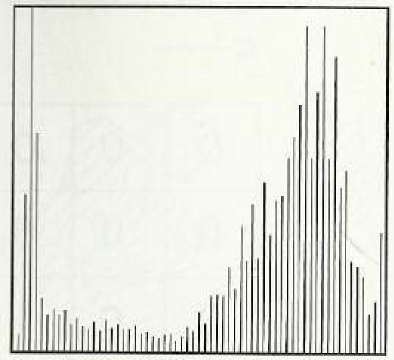
Figure 10: Subimage and its histogram

The ***shape of the histogram*** provides us with information about the nature of the image, or sub image if we are considering an object with the image. For example, a ***very wide*** histogram implies a high contrast, a ***very***

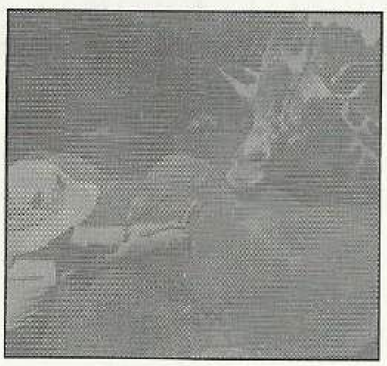
narrow histogram implies a low contrast, a histogram ***skewed toward the right*** implies a bright image, a histogram ***skewed toward the left*** implies a dark image, and a histogram with ***two major peaks***, implies an object that in contrast with the background.



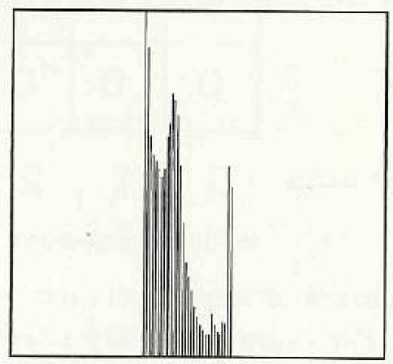
a. Object in contrast with background.



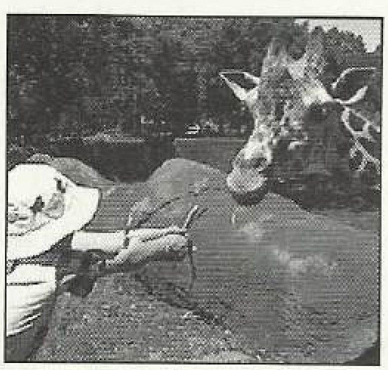
b. Histogram of (a) shows bimodal shape.



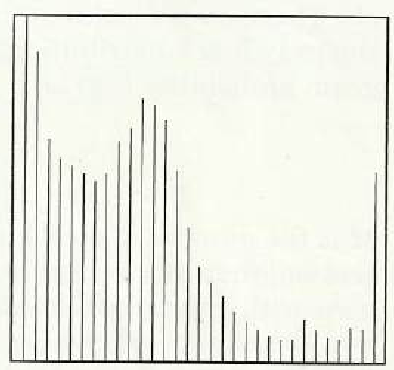
c. Low-contrast image.



d. Histogram of (c) appears clustered.



e. High-contrast image.



f. Histogram of (e) appears spread out

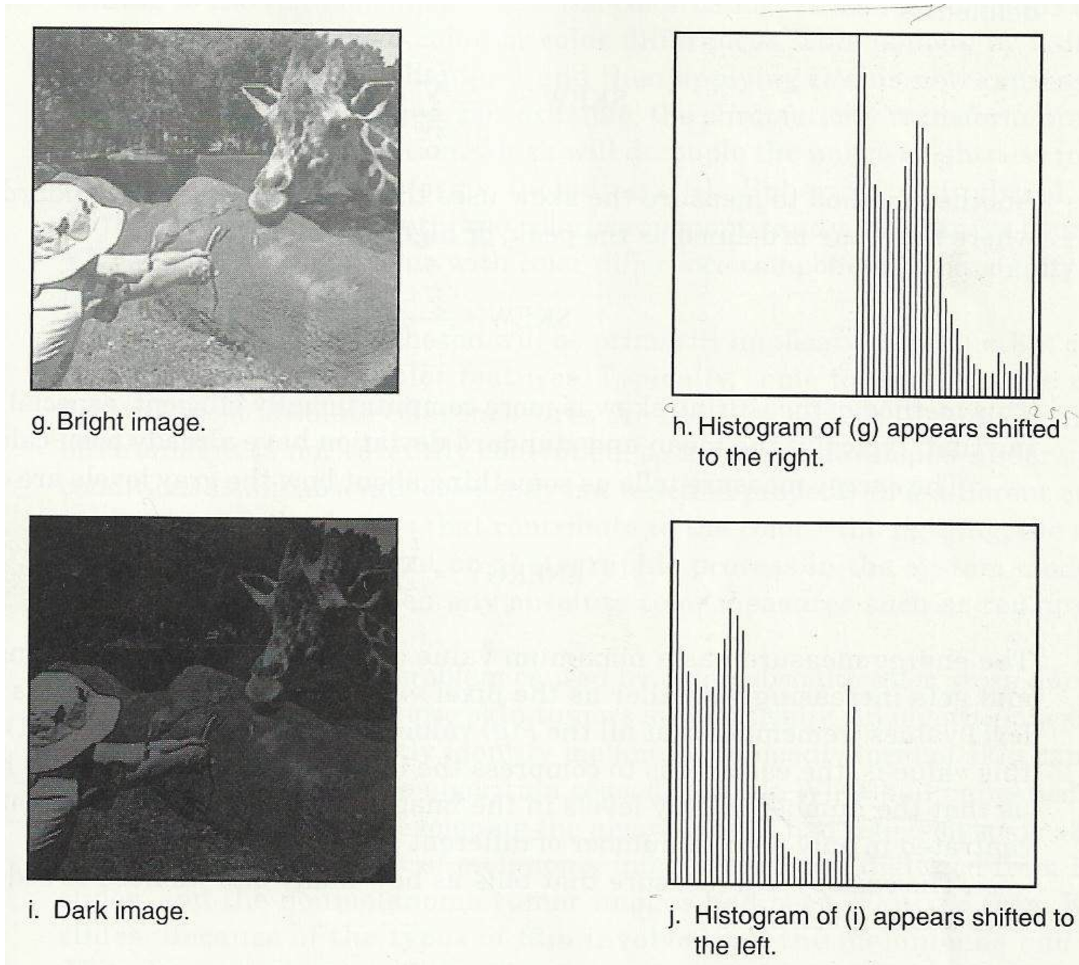


Figure 11 : a variety types of histograms

Probabilistic Histogram

We can ***normalize*** a histogram by ***dividing the counts in each bin*** by the ***total number of pixels*** in the image associated with that histogram. This gave us a table of estimated probabilities. i.e. probability density function (pdf) of the image. Hence, the entry for any gray level tells us the ***likelihood*** of finding that gray level at pixel selected randomly from the image.

However, probabilistic histogram should be used when ***comparing*** the histograms of images with ***different sizes*.**

4.1 Properties and Usage of histogram

One of the ***principle uses*** of the histogram is in the selection of ***threshold*** parameter.

Histogram can be considered as the ***first step*** in ***image matching***, i.e. it is used as a dimensionality reduction technique.

The histogram of an image provides a ***useful indication*** of the relative importance of different gray levels in an image; indeed, it is sometimes possible to ***determine*** whether ***brightness*** or ***contrast adjustment*** is necessary merely by ***examining*** the ***histogram*** and ***not the image*** itself.

The histogram provides sufficient characteristics such as ***invariant to translation and rotation*** of the image, besides ***normalizing the histogram*** leads to achieve invariant properties ***against the scaling*** effect of the image.

When an image is condensed into a histogram, ***all spatial information is discarded*****.** The histogram specifies the number of pixels having each gray level but ***gives no hint*** as to ***where those pixels are located*** within the image. Thus, the histogram is ***unique*** for any particular image, but the ***reverse is not true***. Vastly different images could have identical histograms. Such operations as ***moving objects around within an image*** typically have ***no effect*** on the histogram.

This is evident from figure **12** which shows two very different images that have identical histograms. Although a histogram gives us the frequency distribution of gray levels in an image. It can ***tell us nothing*** about the way in which gray levels are ***distributed spatially***.

Gray level mapping operations affect the histogram of an image in predicable ways. For example, ***adding*** a constant bias to gray levels will ***shift*** a histogram along the gray level axis without changing its shape. ***Multiplication*** of gray levels by a constant gain will ***spread out*** the histogram evenly if $a > 1$, increasing the spacing between occupied bin , or c***ompress*** the histogram it if $a < 1$, which can have the effect of merging bins .

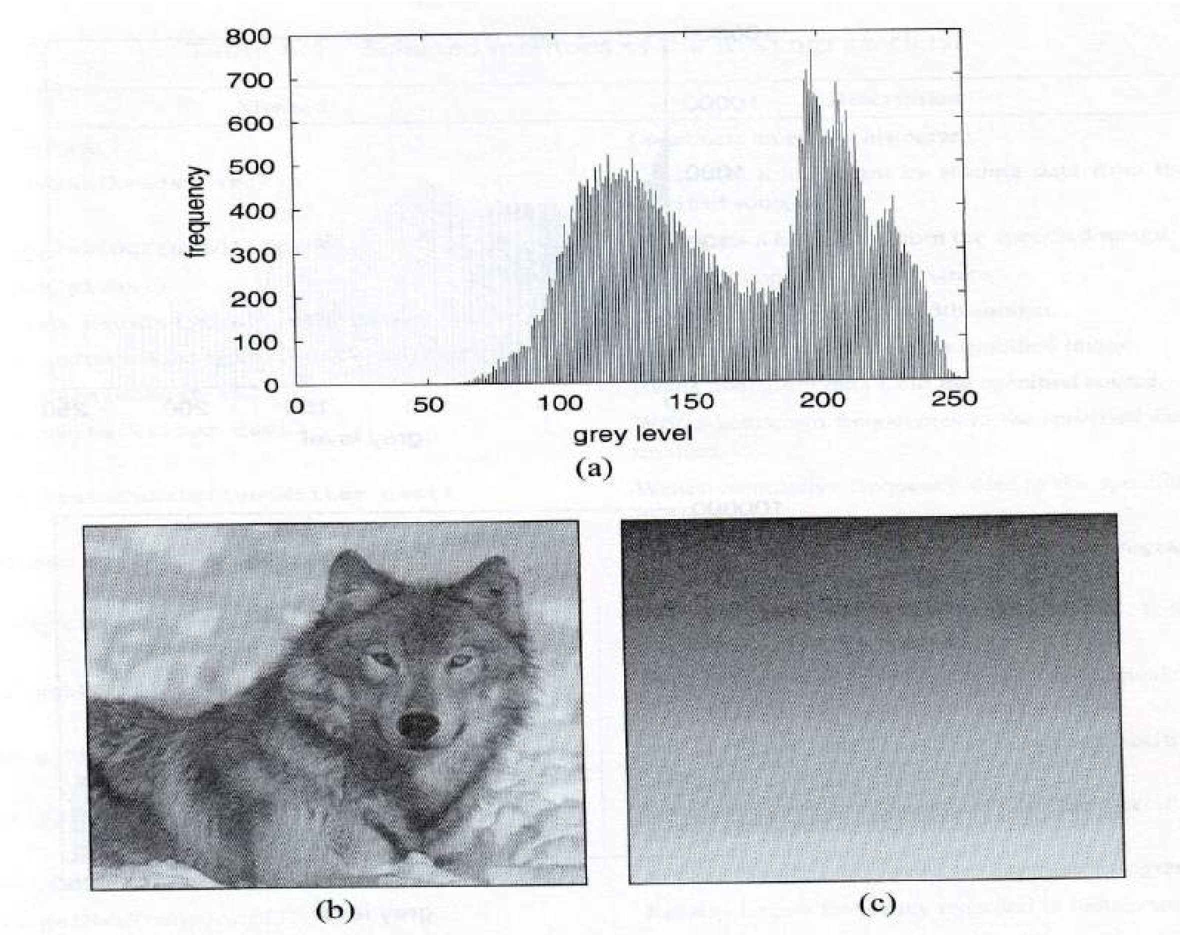


Figure 12: non-uniqueness of a histogram

(a) a histogram (b) an image with (a) as its histogram

(c) a different image with the same histogram

4.2 Histogram modification

An alternate perspective to gray-level modification that performs a similar function is referred to as histogram modification. The gray-level histogram of an image is the distribution of the gray levels in an image. In figure 11 we have see an image and its corresponding histogram. In general a histogram with a ***small-spread*** has a ***low-contrast***. And a histogram with a ***wide spread*** has a ***high contrast***, whereas an image with its histogram clustered at the ***low end*** of the range is ***dark***, and a histogram with the values clustered at the ***high end*** of the range corresponds to a ***bright*** image.

Note The ***main*** effect of the histogram modification operations is on the image histogram itself, but ***it's also effect*** on image contrast (histogram stretch, histogram shrink) and on the image brightness (histogram slide).

The histogram can also be modified by a mapping function, which will either stretch, shrink (compress), or slide the histogram. Histogram stretching and histogram shrinking are forms a gray-level modification, sometimes referred to as histogram scaling. In figure **13** we see a graphical representation of histogram stretch, shrink, and slide.

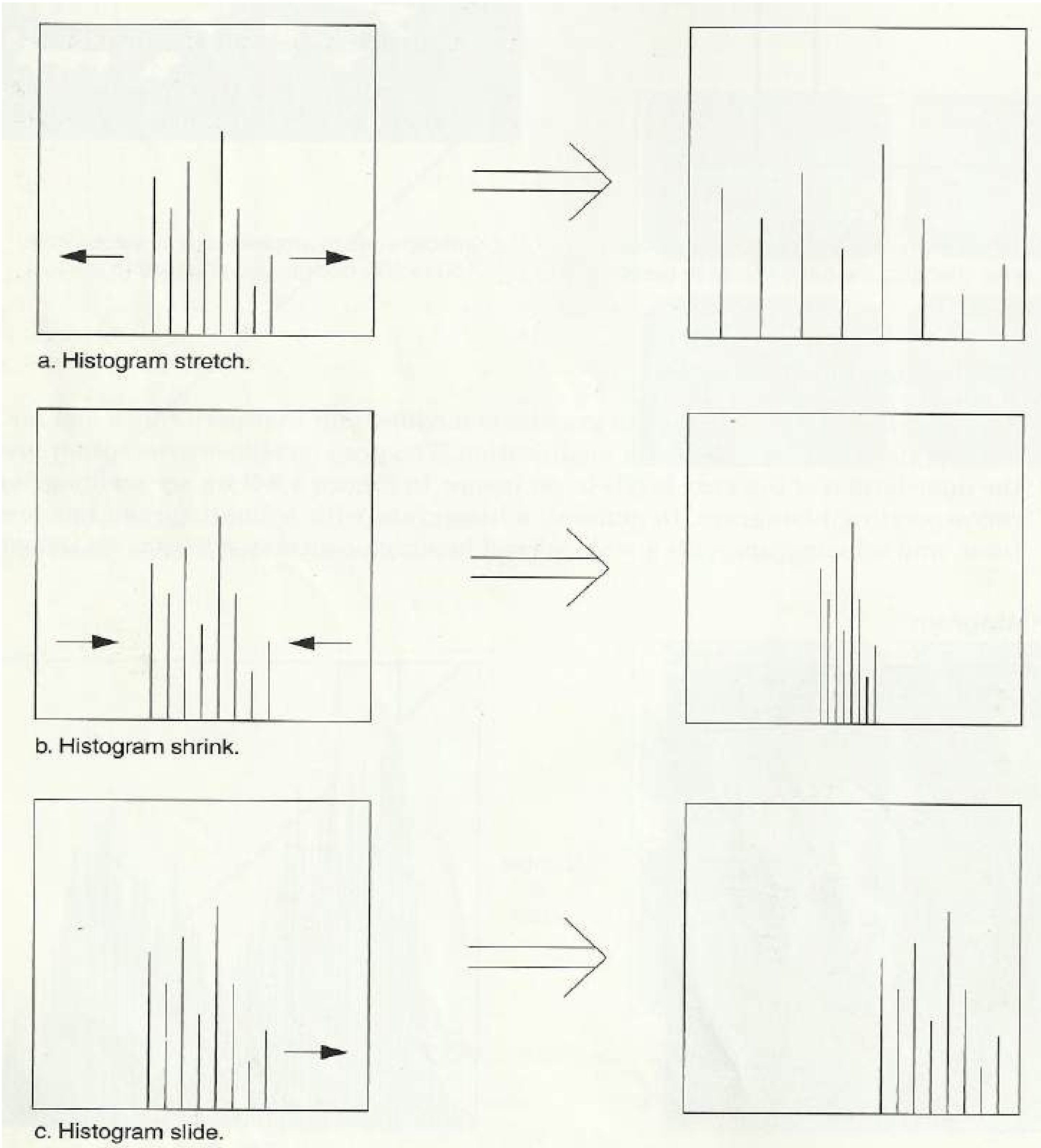


Figure 13 : histogram modification

a. Histogram stretch

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

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

Where : $I(r,c)_{MAX}$ is the largest gray-level value in the image I(r,c)

$I(r,c)_{MIN}$ is the smallest gray-level value in the image I(r,c)

MAX and MIN correspond to the maximum and minimum gray-level values ***desired*** in the stretched histogram (for 8-bit images the typical range is between 0 and 255).

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. ***If a stretch is desired over a smaller range, different MAX and MIN values can be specified.***

In general, histogram stretch will ***increase image contrast***.

ملاحظة : في حال عدم تحديد القيمة العليا والدنيا MIN, MAX في السؤال **فتعتمد القيم الأفتراضية** صفر ، ٢٥٥ أما اذا حددت في السؤال فيجب لألتزام بالقيم المحددة.

Example1:

Apply histogram stretch to the below subimage :

20	90
75	30

Stretch (20) = $[\frac{20-20}{90-20}][255-0] + 0 = 0$

Stretch (90) = $[\frac{90-20}{90-20}][255-0] + 0 = 255$

Stretch (75) = $[\frac{75-20}{90-20}][255-0] + 0 = [\frac{55}{70}]\, 255 = 200.35 = 200$

Stretch (30) = $[\frac{30-20}{90-20}][255-0] + 0 = [\frac{10}{70}]\, 255 = 36.42 = 36$

The resulted sub image is:

0	255
200	36

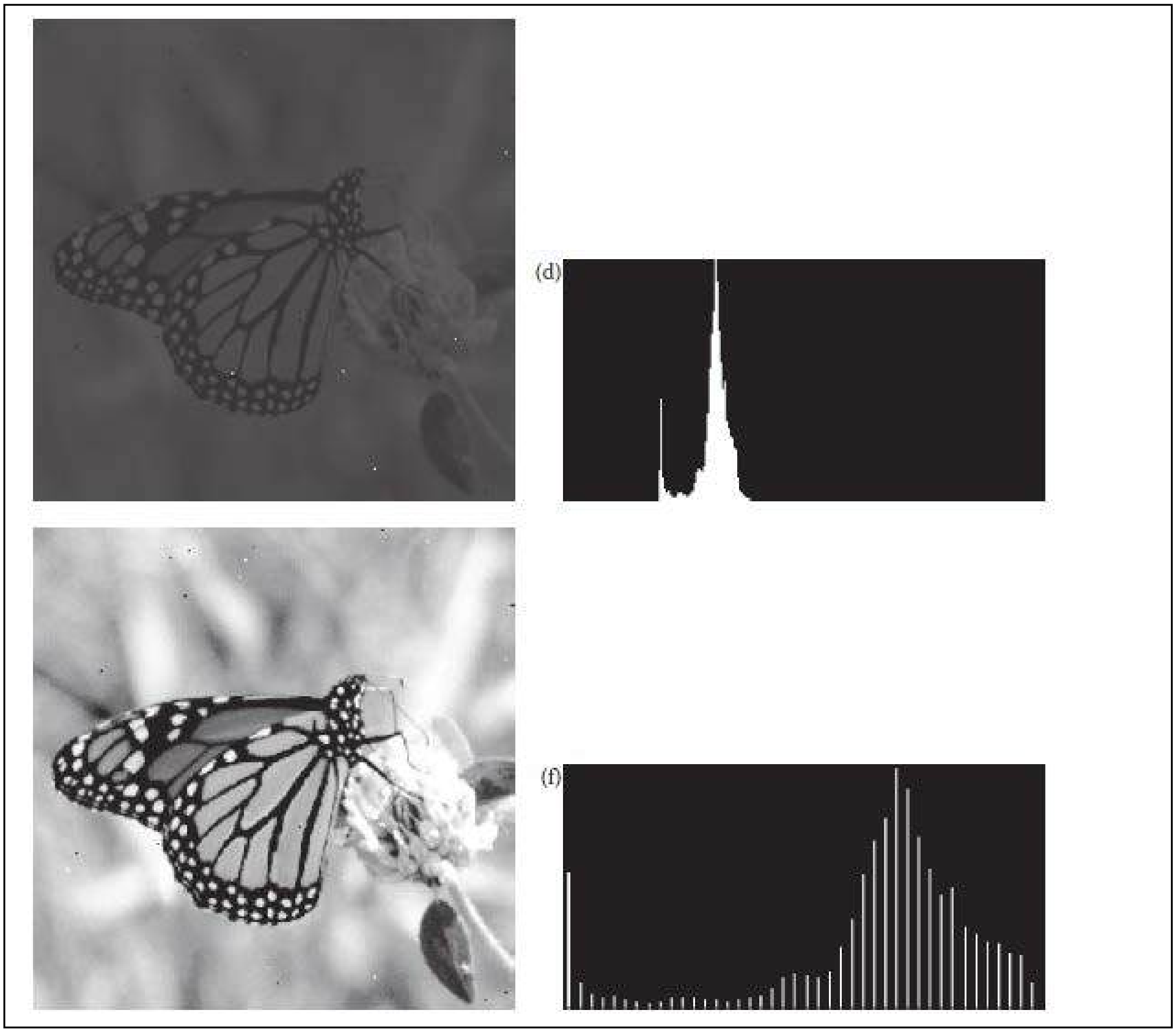


Figure 14: histogram stretch

b. <u>Histogram shrink</u>

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 shrink can be found by the following equation:

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

Where :

$I(r,c)_{MAX}$ is the largest gray-level value in the image I(r,c)

$I(r,c)_{MIN}$ is the smallest gray-level value in the image I(r,c)

$Shrink_{MAX}$ and $Shrink_{MIN}$ correspond to the maximum and minimum gray-level values ***desired*** in the compressed histogram.

In general, this process produces an image of ***reduced contrast*** and may not seem to be useful as an image enhancement tool.

ملاحظة :

في السؤال و يجب الألتزام بالقيم المحددة MAX, MIN **يجب** تحديد قيمة Shrinkفي حالة

Example2 :

Apply histogram shrink to the below subimage where the desired gray level values for the compressed histogram are 40, 80:

20	90
75	30

Sol

Shrink (20) = $[\frac{80-40}{90-20}][20-20] + 40 = 40$

Shrink (90) = $[\frac{80-40}{90-20}][90-20] + 40 = [\frac{40}{70}][70] + 40 = 80$

Shrink (75) = $[\frac{80-40}{90-20}][75-20] + 40 = [\frac{40}{70}][55] + 40 = 71.42=71$

Shrink (30) = $[\frac{80-40}{90-20}][30-20] + 40 = [\frac{40}{70}][10] + 40 = 45.71=46$

The resulted sub image is :

40	80
71	46

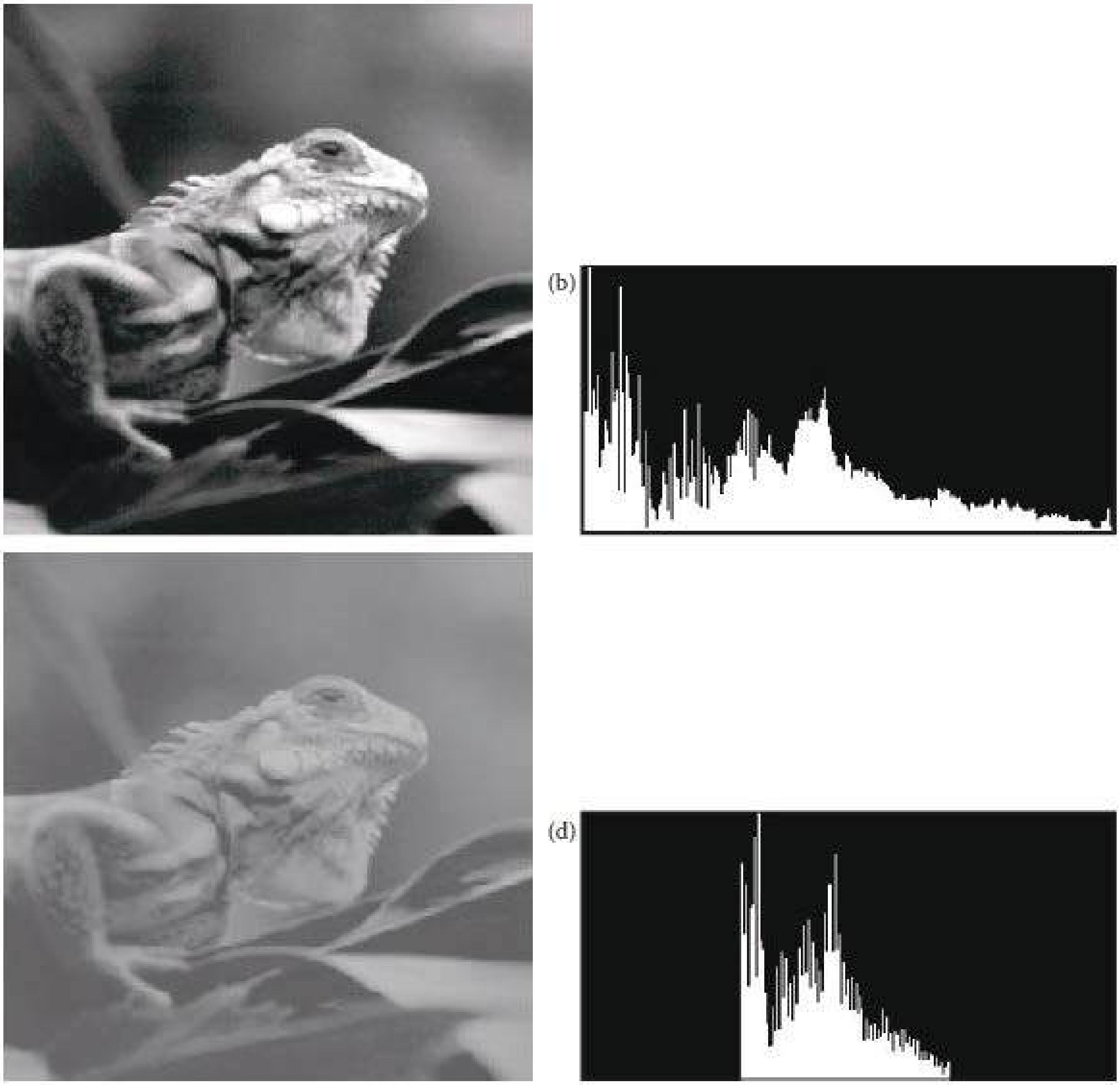


Figure 15: histogram shrink

c. Histogram Slide

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

$$Slide(\ I(r,c)\ ) = I(r,c) + OFFSET$$

Where OFFSET value is the amount to slide the histogram**.**

In this equation we assume that any values slide past the minimum and maximum value will be clipped to the respective minimum or maximum. A **positive** OFFSET value will **increase** the overall **brightness**, whereas a **negative** OFFSET will create a **darker** image. Figure 15 shows a dark image that has been brightened by a histogram slide with a positive OFFSET value.

Example3

Make the below subimage brighter by 100 gray level value.

20	90
75	30

Sol

Slide (20) = 20+100 = 120

Slide (90) = 90+100 = 190

Slide (75) = 75+100 = 175

Slide (30) = 30+100 = 130

The resulted sub image is :

120	190
175	130

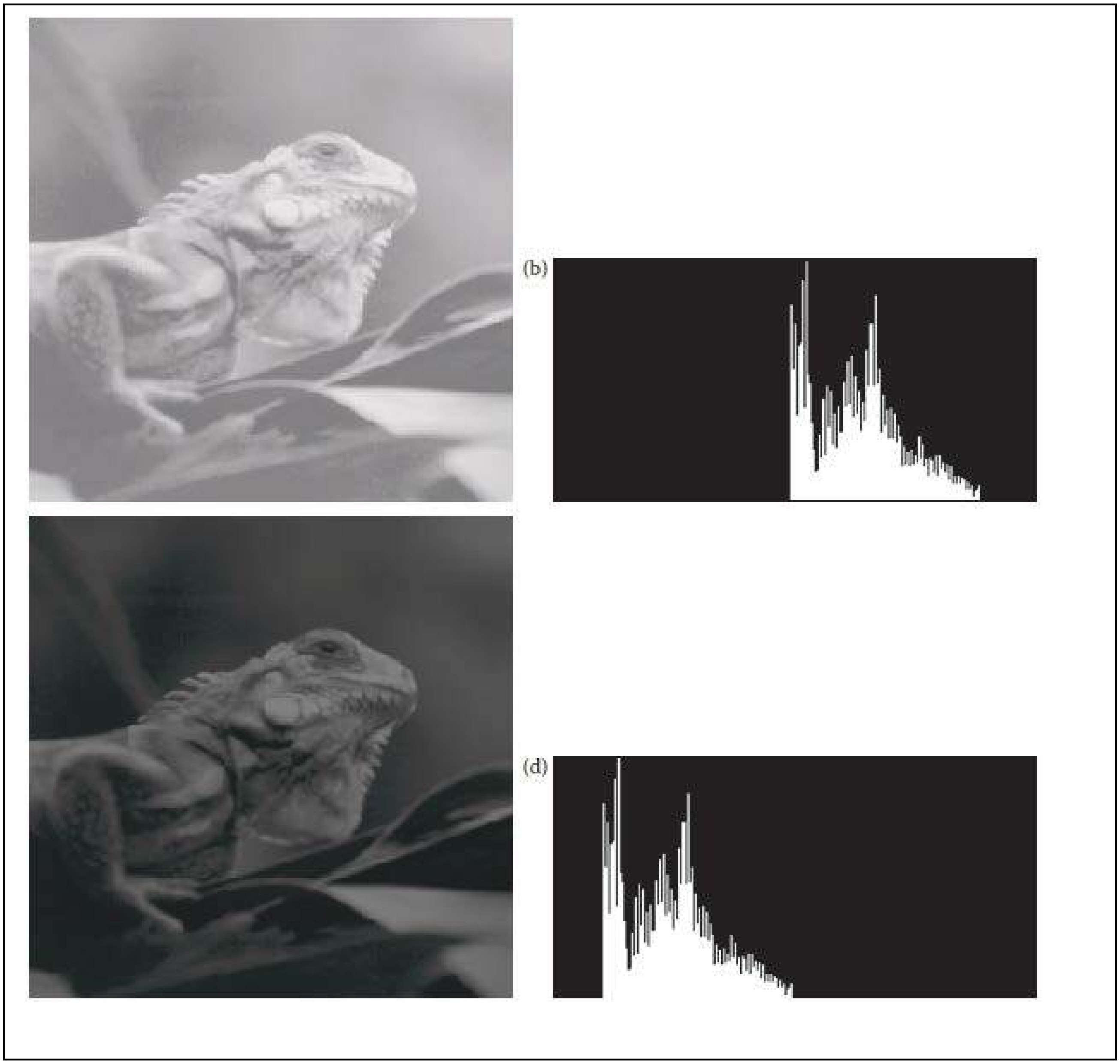


Figure 16: Histogram Slide

d. Histogram Equalization

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

Histogram equalization, ***redistribute*** gray levels in an attempt to ***flatten*** the frequency distribution. ***More gray levels are allocated where there are most pixels, fewer gray levels where there are fewer pixels.*** This tends to ***increase contrast*** in the most heavily populated regions of the image.

If we are to ***increase contrast*** for the ***most frequently*** occurring gray level range and ***reduce contrast*** in the ***less popular part*** of the gray level rang, then we need a mapping function which has a steep slope ($a > 1$) at gray levels that occur frequently, and a gentle slope ($a < 1$) at unpopular grey levels. The cumulative histogram of the image has these properties.

Indeed, the mapping function we need is obtained simply by rescaling the cumulative histogram so that its values lie in the range 0-255.The algorithm below shows how this works in practice. From the histogram of the image, we determine the cumulative histogram, C, rescaling the values as we go so that they occupy an 8-bit range. In this way, C becomes a look-up table that can be subsequently applied to the image in order to carry out equalization.

Algorithm 4 : Histogram Equalization

Compute a scaling factor $\alpha = 255$ / number of pixels.

Calculate histogram using **algorithm 3**

C[i] = α * histogram [0]

For all remaining gray levels, I , do

C[i] = C[i-1] + α * histogram [i]

End for

For all pixel coordinate, x and y, do

g (x,y) = C[f (x,y)]

End for

Histogram Equalization is used widely in image processing, mainly because it is a ***completely automatic*** technique, with ***no parameter to set***. At times, it can improve our ability to interpret an image dramatically.

Histogram equalization may ***not*** always ***provide the desired effect*** because its goal is fixed – to distribute the grey level value as evenly as possible. However, it is difficult to predict how beneficial equalization will be for any given image; in fact, it may not be of any use at all. This is because the ***improvement in contrast is optimal statistically, rather than perceptually***. In images with narrow histograms and relatively few grey levels, a massive increase in contrast due to histogram equalization can have the adverse effect of reducing perceived image quality.

Steps of Histogram Equalization

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

1. Find the **running sum** of the histogram.
2. **Normalize** the values from step 1 by **dividing** by the total number of pixels.
3. **Multiply** the values from step 2 by the **maximum** gray level value and **round**.
4. **Map** the gray levels value to the results from step 3 using a one-to-one correspondence.

Example

Assume we have an image with ***3 bits/pixels***, so the possible range of the pixels values is 0 to 7. We have an image with the following histogram

Gray levels value	**number of pixels (Histogram values)**
0	10
1	8
2	9
3	2
4	14
5	1
6	5
7	2

STEP 1:_Create a **running sum** of the histogram value. This means that the first value is 10, the second is 10+8=18, next is 10+8+9=27, and so on. Here we get 10, 18, 27, 29, 43, 44, 49, 51.

STEP 2: **Normalize** by dividing by the total number of pixels. The total number of pixels is 10 +8+9+2+14+1+5+2 = 51 (note that this is the last number from step 1), so we get 10/51, 18/51, 27/51, 29/51, 43/51, 44/51, 49/51, 51/51.

STEP 3 : **Multiply** these values by the **maximum gray-level** values, in this case 7, and then **round** the result to the closest 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 step 3 by a one-to-one correspondence. This is done as follows

<table>
<tr><th>Original Gray level value</th><th>Histogram Equalized Gray level values</th><th>Number of pixels</th></tr>
<tr><td>0</td><td>1</td><td>10 (frequency of 0)</td></tr>
<tr><td>1</td><td>2</td><td>8 (frequency of 1)</td></tr>
<tr><td>2</td><td>4</td><td rowspan="2">11 (frequency of 2 + 3)</td></tr>
<tr><td>3</td><td>4</td></tr>
<tr><td>4</td><td>6</td><td rowspan="2">15 (frequency of 4 + 5)</td></tr>
<tr><td>5</td><td>6</td></tr>
<tr><td>6</td><td>7</td><td rowspan="2">7 (frequency of 6 + 7)</td></tr>
<tr><td>7</td><td>7</td></tr>
</table>

In the below figure we see the original histogram and resulting histogram (equalized histogram). Although the result is not flat, it is closer to being flat than the original histogram.

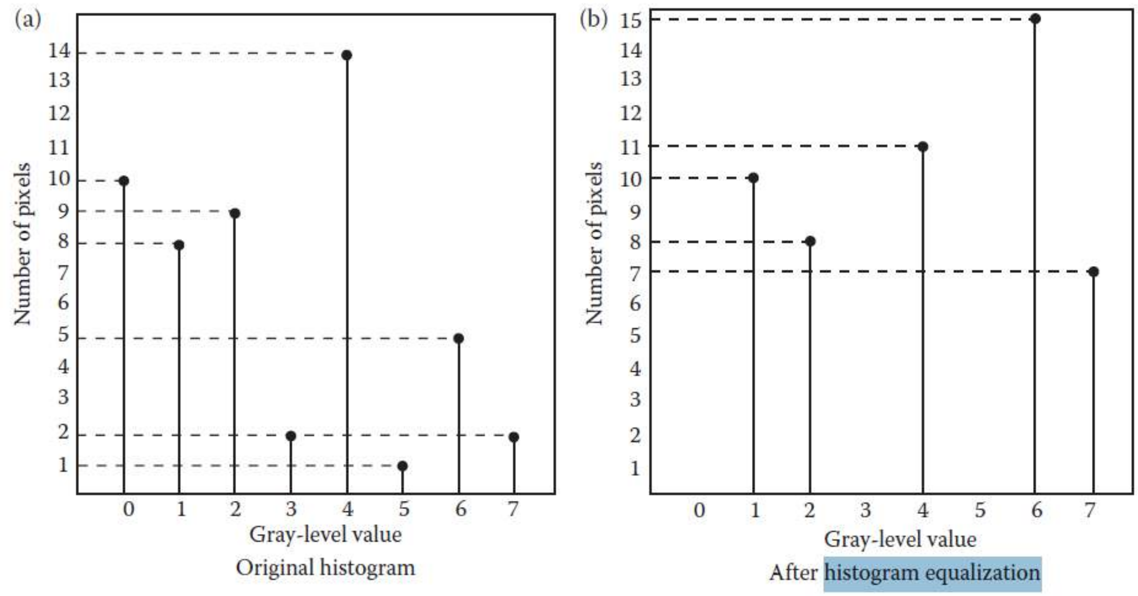


Figure 17 Histogram Equalization

Histogram equalization of a digital image will not typically provide a histogram that is perfectly flat, but will ***make it as flat as possible***.

The below figures shows the result of equalizing the histogram of two images with very poor contrast.

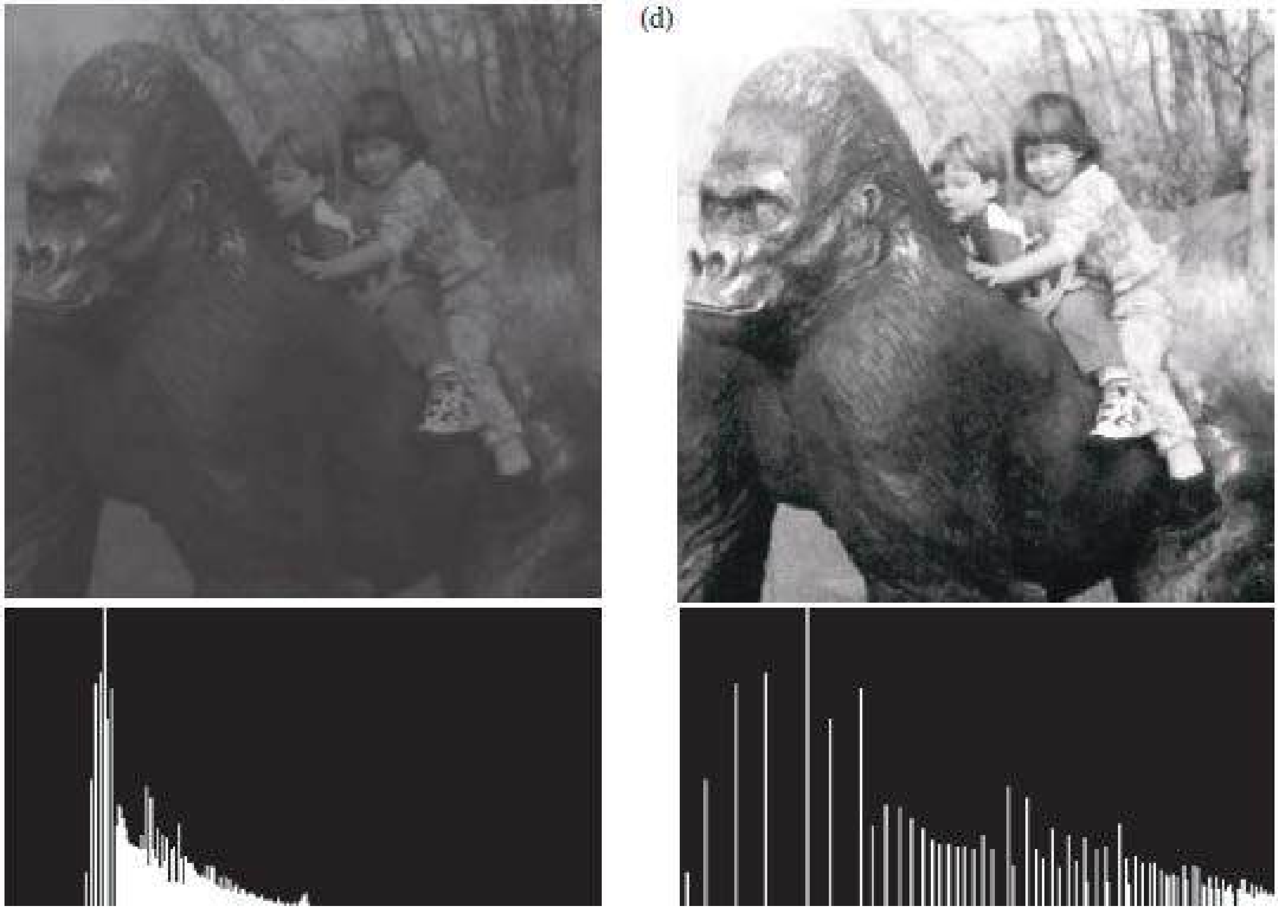


Figure 18 Histogram Equalization

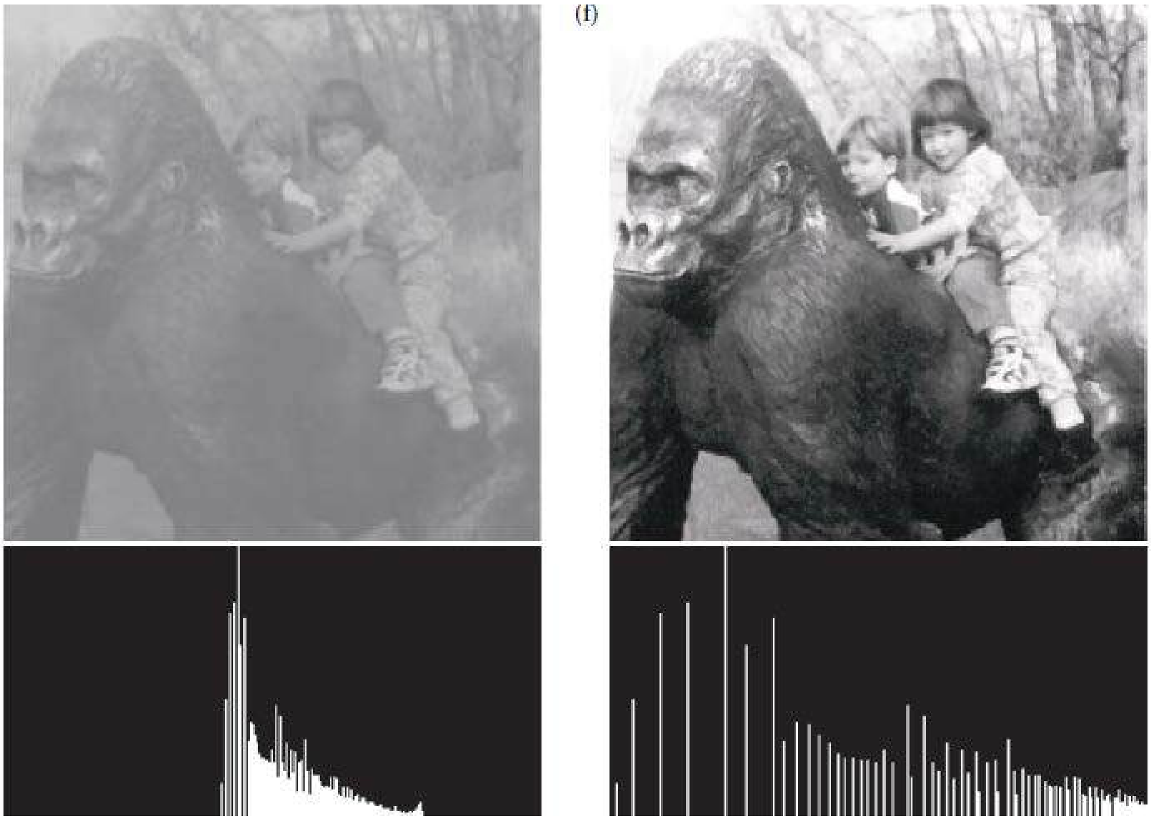


Figure 18 Histogram Equalization (continue)

Introduction to Digital Video

1. Preview

The term video used to represent a stream of images and audio. In the last few years, there has been a growing increase in the use of digital video due to the expansion in both the use of multimedia technology and Internet technology. The video is a rich source of information. It provides visual information about scenes.

Video has become a part of our everyday life, think of television Broadcast for example. It is the most effective medium for ***capturing*** the ***events*** in the real world around us. It is also the most dramatic medium as it ***combines*** both photo-realistic images and sounds. Combining the advantages of video and computers will broaden the scope of information that a computer can process.

The video is composed of ***spatial*** (pixels in frame) and ***temporal*** (frames in time interval) components. These components provide a compact description of the video data. The video data can be transformed from a sequential frame-based representation, in which this common scene information is distributed over many frames, into a single common scene-based representation to which each frame can be directly related. This representation then allows direct and immediate access to the scene information, such as static locations and dynamically moving objects.

2. Video File Formats

There are different layers of video transmission and storage, each with its own set of formats to choose from.

For transmission, there is a physical connector and signal protocol (video connection standard). Many analog and digital recording formats are in use, and digital video clips can also be stored on a computer file system as files which have their own formats. In addition to the physical format used by the data storage device or transmission medium, the stream of ones and zeros that is sent must be in a particular digital "video encoding", of which a number are available

FILE FORMAT	EXTENSION	MORE INFORMATION
Adobe Flash Media	.swf	**Flash Video** This file format is generally used to deliver video over the Internet using th Adobe Flash Player.
Windows Media file	.asf	**Advanced Streaming Format** This file format stores synchronized multimedia data and an be to stream audio and video content, images, and script commands over a network.
Windows Video file	.avi	**Audio Video Interleave** This is a multimedia file format for storing sound and moving p tures i Microsoft Resource Interchange File Format (RIFF) format. It is one of the most common rmats because audio or video content that is compressed with a wide variety of codecs can be ored ir .avi file.
Movie file	.mpg or .mpeg	**Moving Picture Experts Group** This is an evolving set of standards for video and audio mpres developed by the Moving Picture Experts Group. This file format was designed specificall for use Video-CD and CD-i media.
Windows Media Video file	.wmv	**Windows Media Video** This file format compresses audio and video by using the Wind ws Me Video codec, a tightly compressed format that requires a minimal amount of storage spa e on yo computer's hard disk.

3. Digital Video

Digital video is a large subject that draws upon an equally large number of technologies such as television broadcast, video-phone, teleconferencing, satellite observations, and medical imaging. Video is the most effective medium for ***capturing the events*** in the real world. It becoming more popular and accessible through the various media technology advances which enable users to capture, manipulate and store video data in efficient and inexpensive ways. With the increasingly efficient ***compression formats*** and ease of ***integrating videos in web pages***, more people are able to enjoy production and publishing movies in the digital world .

Since video is dynamic, the visual content evolves with time and generally contains moving objects. The information contained in video is ***much richer*** than is contained in still images, since our world is constantly changing according to the movement of people, animals, vehicles, and other objects .

Digital video is ordinarily a function of ***three dimensions***, two in ***space*** and one in ***time***, as depicted in Figure (1). Because of this, digital video processing is data intensive: significant bandwidth, computational, and storage resources are required to handle video streams in digital format. Digital video comprises a series of digital images displayed in rapid succession at a ***constant rate***. In the context of video, these images are called ***frames***. The rate at which frames are displayed is measured in frames per second (FPS). Each frame is a ***snapshot*** of a moment in time of the motion video data, and is very similar to a still image. It comprises a raster of pixels . If it has a width of pixels and a height of pixels, the frame size is width × height.

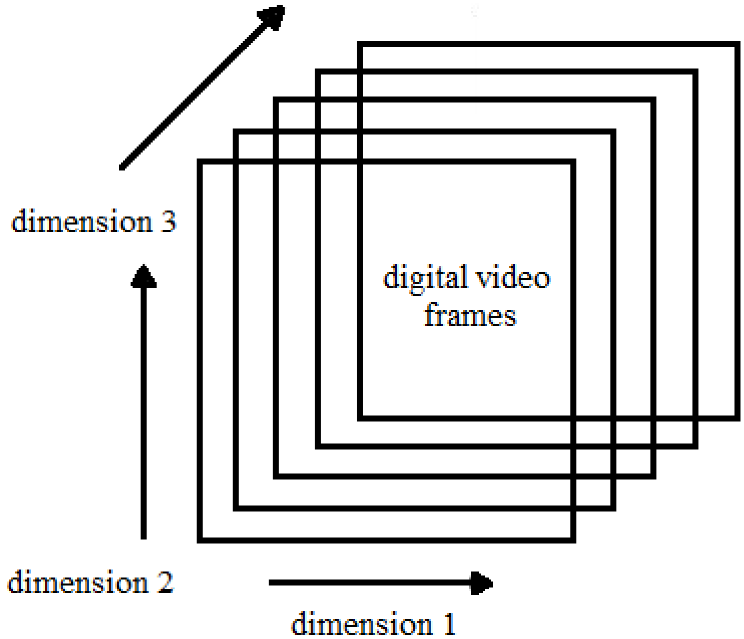


Figure (1) : the dimensionality of video

3.1 Basic Digital Video Concepts

The following is a list of aspects of digital video that can be manipulated with standard video-editing software. It is important to be familiar with these terms so you can create video optimized for web delivery.

Movie Length

It's a simple principle—limiting the length of your video clip will limit its file size. Videos longer than a minute or two may create prohibitive download times. If you must serve longer videos, consider one of the ***streaming video*** solutions.

Frame Size

Obviously, the size of the frame will have an impact on the size of the file. "***Full-screen***" video is 640×480 pixels. The amount of data required to deliver that size image would be prohibitive for most web applications. The most common frame size for ***web video*** is 160×120 pixels. Some producers will go as small as 120×90 pixels. It is not recommended that you use a frame size larger than 320×240 with current technology. Actual size limits depend mostly on ***CPU power*** and ***bandwidth*** of your Internet link.

Frame Rate

The frame rate is measured in number of frames per second (fps). ***Standard TV-quality*** video uses a frame rate of 25-30 frames per second to create the effect of smooth movement. For the ***Web***, a frame rate of 15 or even 10 fps is more appropriate, and is still capable of producing fairly smooth video playback. For “talking head” and other low-motion subjects, even lower frame rates may be useful. Commercial Internet broadcasts are routinely done at 0.5, 0.25, or even 0.05 frames per second (resulting in a slideshow effect rather than moving video).

Data Rate (bit rate)

This is the rate at which data must be transferred for the video to play smoothly without interruption. The data rate (also called " bit rate") for a movie is measured in kilobytes per second (K / sec or K bps). It can be calculated by dividing the size of the file (in K) by the length of the movie (in second). So, for example , a highly compressed movie that is 1900 K (1.9 MB) and 40 seconds long has a data rate of 47.5 K/sec.

For streaming media in particular, a file's data rate is more important than its total size. this is because the total bandwidth available for delivery may be severely limited , particularly over a dial-up connection.

The color of a pixel is represented by a fixed number of bits. The more bits the more subtle variations of colors can be reproduced . This is called the ***color depth (CD)*** of the video. The most important properties of video are the bit rate and the video size. The equations of these two properties and other properties are:

PF = W * H .... (1)

BF = W * H * CD .... (2)

PS = W * H * FPS .... (3)

BR= BF * FPS .... (4)

VS = BR * D .... (5)

Where W is the width of the frame, H is the height of the frame, PF is the pixel per frame , CD is the color depth, BF is the bit per frame, FPS is the frame per second, PS is the pixel per second, BR is the bit rate, D is the duration of video in seconds, VS is the video size.

Ex1

a video have a duration (T) of 1 hour (3600*sec*), a frame size of 640×480 (W×H) at a color depth of 24 bits and a frame rate of 25 fps.

This example video has the following properties:

pixels per frame (**PF**) = 640 * 480 = 307,200 Pixel

bits per frame (**BF**) = 307,200 * 24 = 7,372,800 = 7.37 *Mbits*

pixel per second (**PS**) = 307,200 * 25 = 7680000 Pixel/sec.

bit rate (**BR**) = 7.37 * 25 = 184.25 *Mbits/sec*

video size (**VS**) = 184*Mbits/sec * 3600sec = 662,400Mbits = 82,800Mbytes .*

Ex2

a video uses PAL standard, have a duration (T) of 0.5 hour (1800 sec) , its color depth of 24 bits, if you know that the video required 630,000 Mbit to be stored on the hard disk find : BR, BF, PF.

Sol

$VS = BR \times D$

$BR = VS \backslash D = 630000 \backslash 1800 = 350$ Mbit/Sec

$BF = BR \backslash FPS = 350 \backslash 25 = 14$ Mbit/frame $= 14 \times 10^6$ bit/frame

$BF = W \times H \times CD$

$14 \times 10^6 = W \times H \times 24$

$W \times H = 14 \times 10^6 \backslash 24 = 583333$ Pixel.

$PF = W \times H = 583333$ Pixel/frame.

In real-time video, the playback rate is 30 frames per second. This is the minimum rate necessary for the human eye to successfully blend each video frame together into a continuous, smoothly moving image.

Advantages of Digital Video

The advantages of digital representation for video are many. It permits

1. Storing video on digital devices or in memory, ready to be processed (noise removal, cut and paste, and so on) and integrated into various multimedia applications.

2. Ease of sharing and storage.
3. Direct access, which makes nonlinear video editing simple.
4. Repeated recording without degradation of image quality.
5. Ease of encryption and better tolerance to channel noise.
6. Efficient compression.

4. Analog Video

Most TV is still send and received as an ***analog signal***. Once the electrical signals are received, we assume that brightness is at least a monotonic function of voltage.

In TV and in monitors and multimedia standards, an ***interlaced*** scanning, system is used. Here, the Odd numbered lines are traced first, then the even numbered lines, these results in "odd" and "even" fields- two fields ***make up one frame.***

For displaying a video on TV and old monitors interlacing technique is used; ***interlacing is the process of dividing each video frame into two halves one contains the odd liens only and the other contains the even lines so each frame will be displayed two times***, if a video has 30 frames/sec using interlacing each frame will be displayed twice first display the odd lines and then the even lines will be displayed, this means that the video will be displayed using 60 frames instead of 30 frames, ***interlacing was introduced because the huge size of video file and in the beginning it was hard to send huge amount of information in one frame quickly without suffering from flickering***.

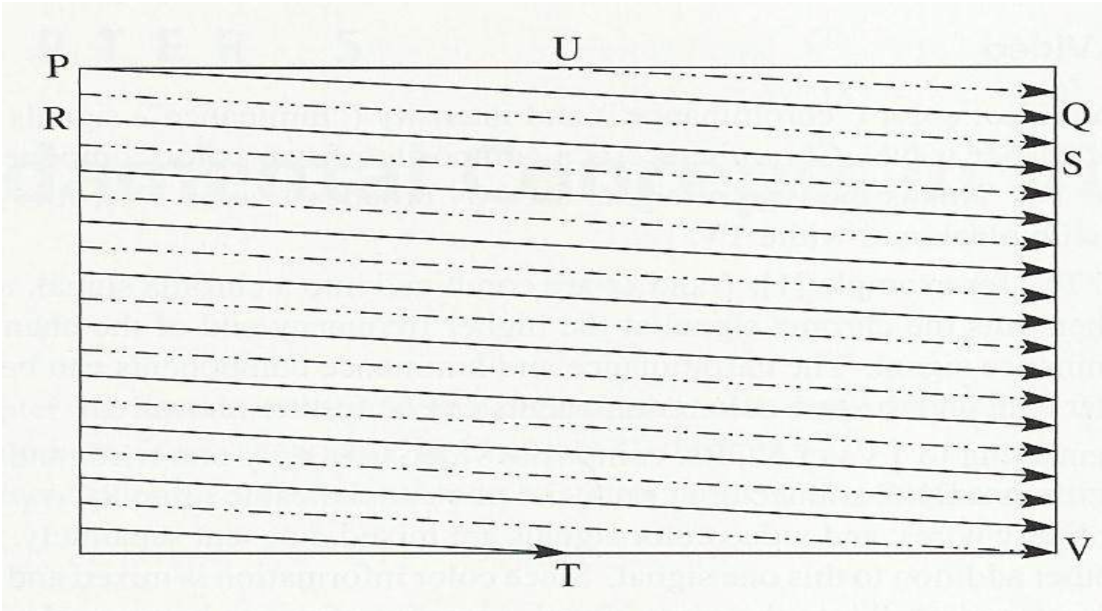


Figure (2): Interlaced Raster Scan

4.1 NTSC video

The NTSC standard is mostly used in ***North America and Japan***. It uses a familiar ***4:3 aspect ratio*** (i.e. the ratio of picture width to height) and ***525 scan lines*** per frame at ***30 frames per second***.

More exactly, NTSC uses 29.97 fps – or, in other words, 33.37 msec per frame. NTSC follows the interlaced scanning system, and each frame is divided into ***two fields***, with ***262.5 lines/field***. Thus the horizontal sweep frequency is 525× 29.97 ≈ ***15,734 lines/sec***, so that each line is swept out in 1/15,734 sec ≈ ***63.6 µsec***.

NTSC uses **YIQ** color model. NTSC assigned a ***bandwidth of 4.2 MHz to Y*** but only **1.6** MHz to **I** and **0.6** MHz to **Q**, due to the ***humans' insensitivity to color detail***s (high frequency color changes).

4.2 PAL video

PAL (**P**hase **A**lternating **L**ine) is a TV standard originally invented by ***German*** Scientists. It uses 625 scan line per frame, at 25 frame per second (or **40** msec/frame), with 4**:3** aspect ratio and interlaced field. Its broadcast TV signals are also used in ***composite video***. This important standard is widely used in Western Europe, China, India and many other parts of the world.

PAL uses the **YUV** color model with **8** MHz channel bandwidth, ***allocating a bandwidth of 5.5 MHz to Y*** and **1.8** MHz each to **U** and **V**.

4.3 SECAM video

SECAM, which was invented by the ***French***, is the third major broadcast TV standard. SECAM stand for System **E**lectronique **C**ouleur **A**vec **M**emoire. SECAM also uses **625** scan lines per frame, at **25** frames per second, with **4:3** aspect ratio and interlaced field.

SECAM and PAL are similar, ***differing*** slightly in their ***color coding*** scheme.

<u>Ex</u>

Assume you have a TV uses PAL TV standard, calculate:

1. The number of lines does this TV emit per second.
2. The time (in msec) this TV required to emitting one frame.

Sol

1. PAL TV's uses 625 lines per frame, at 25 frames per second.

   Number of lines per second = 625 × 25 = 15,625 lines/second.

2. PAL TV's emit 25 frames per second.

   It emits 25 frames per 1,000 msec.

   Time required to emit one frame=1000 / 25 = 40 msec/frame.

Table 1 gives a comparison of the three major analog broadcast TV systems.

Table 1 : comparison of analog broadcast TV system.

TV system	Frame rate (fps)	Number of scan lines	Total channel width (MHz)	Bandwidth allocation (MHz)		
				Y	*I* or *U*	*Q* or *V*
NTSC	29.97	525	6.0	4.2	1.6	0.6
PAL	25	625	8.0	5.5	1.8	1.8
SECAM	25	625	8.0	6.0	2.0	2.0

5. Video Streaming

Video had become an important data type in communication and entertainment, in the beginning videos were transmitted in analog form before developing digital integrated circuits and the increase of using personal computers and the internet.

Since video file size is huge, then the need to some data reduction method appears, video compression is the process of reducing video data to be stored on some medium or transmitted over the internet, quality is another important aspect for videos after being transmitted or compressed in order to the client has a good watching experience .

Downloading large multimedia files such as video conferencing and surgical operations very quickly is not an easy task, because downloading such video requires a very fast access which typically not all users have.

Video streaming technology refers to transmitting video over network connection directly from the source in ***real time***. In streaming, the end user does not need to wait for video download to finish; "streaming" video will start after a few seconds upon receiving the video frames. ***Technically, the video frames are still "downloading" but the end user does not need to wait before starting to watch.***

The streaming process for video requires : **(1)** large sever to store VoD and power full processors to reduce latency for live video , in addition to the **(2)** available bandwidth of the network that controls the quality of delivered videos.

Studies refers that video streaming over the internet takes about 50% of internet traffic in the peak time and this ratio will be grow in the next few years.

When videos streamed from the server to the client it can be delivered in three ways:

1. **Video delivery as file download**

   One of the approaches of video delivery on the internet is a file download, downloading video is similar to file download, in this case the user must ***wait until all the video file downloaded to his device before he will be able to display it***, this approach usually take a ***long time*** and high network ***latency*** because the size of video files is usually large.

2. **Video delivery as stream of chunks**

    Video in this approach is divided to chunks ***each chunk contains N seconds*** of the video, these chunks can simultaneously streamed and played back in reverse file download approach, the delay time here named ***pre-roll which it is the time between star delivery of video chunks and play back of video on screen***.

3. **Video streaming as a sequence of images**

    Video file can be streamed to the client as a sequence of images or frames ***each frame in the video file must be delivered, decoded and played in its playback time***, otherwise it will be useless if it received out of order and it will be neglected.

6. Video Compression

Video compression is the method of reducing video file size by eliminating some of unnecessary video data to be suitable for storing on some medium or to the transmission to a remote location.

Since video file size is **huge**, then the need to some data reduction method appears, video compression is the process of reducing video data to be stored on some medium or transmitted over the internet. Quality is another important aspect for videos after being transmitted or compressed in order to the client has a good watching experience.

The **widespread of videos on the internet** in addition to the **large size of the video** files and the **limitations of the networks like the bandwidth** fluctuation in the internet **make video compression an important tool** for video communication and video streaming to accomplish fast video streaming with appropriate quality.

There are two main types of compression: lossless video file compression and lossy video file compression.

The **lossless** compression methods compress the video frames in way they can be ***retrieved exactly like the original frames no data lost*** during the coding and decoding these methods usually ***don't give a high compression*** ratio like Dictionary compression techniques.

On the other hand, the **lossy** compression methods ***give a high compression ratio*** but in return ***video frames losses some of their data*** and there is no way to retrieve the lost data back and in result the decoded frames are ***not identical to the original frames and loss some of their quality*** like MPEG and some types of Wavelet compression techniques.

Analog video compression achieved by exploiting the similarity or redundancy in its signal.

In digital videos, The Moving Picture Experts Group (MPEG) method is used to compress video. In principle, a motion picture is a rapid flow of a set of frames, where each frame is an image. In other words, a frame is a spatial combination of pixels, and a video is a temporal combination of frames that are sent one after another. Compressing video, then, means ***spatially*** compressing each frame and ***temporally*** compressing a set of frames. Hence, in digital videos there are two ways of compressing video file:

1. **Spatial compression**: video file is a group of images and in this type of compression each image (or frame) compressed separately from other images in the video, compression is done by regular image compression methods like Wavelet or JPEG compression techniques.
2. **Temporal compression**: In temporal compression ***redundant frames are removed***. In this type of compression each group of pictures "Frames" (GoP) in the video file are compressed together, this method of compression produce higher compression rate and its now used by all new codecs nowadays.

When we watch television, we receive 50 frames per second. However, most of the consecutive frames are almost the same. For example, when **someone is talking**, most of the frame **is the same** as the previous one **except** for the segment of the frame **around the lips**, which changes from one frame to another.

To temporally compress data, the MPEG method first divides frames into three categories: I-frames, P-frames, and B-frames.

1. I-frames. An intra frame (I-frame) is an **independent** frame that is not related to any other frame (not to the frame sent before or to the frame sent after). They are **present at regular intervals** (e.g., every **ninth** frame is an I-frame). An I-frame must appear periodically to handle some sudden change in the frame that the previous and following frames cannot show. I-frames are **independent** of other frames and cannot be **constructed** from other frames.

2. P-frames. A predicted frame (P-frame) ***is related to the preceding I-frame or P-frame***. In other words, each ***P-frame contains only the changes (difference) from the preceding frame***. The changes, however, cannot cover a big segment. For example, for a fast moving object, the new changes may not be recorded in a P-frame. P-frames can be constructed only from previous I- or P-frames. P-frames carry **much less information** than other frame types and carry even fewer bits after compression.

3. B-frames. A bidirectional frame (B-frame) ***is related to the preceding and following I-frame or P-frame***. In other words, each B-frame is relative to the past and the future. Note that a ***B-frame is never related to another B-frame***.

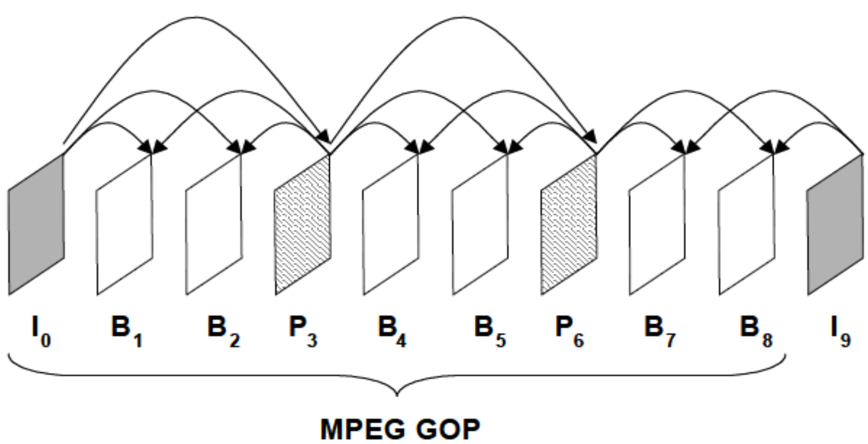


Figure 7: MPEG group of images [ALT02]

The MPEG compression technique uses temporal or block compression, it divides each video frame to 8*8 blocks and start compress the frames according to their type, it ***compresses*** the

Predictive and Bidirectional frames using motion estimation in which is a motion vector contains the difference between the frames or save the motion of objects and ***keeps*** the **Intra-frame** **un compressed** ***(or it could be compressed using image compression techniques like JPEG or Wavelet methods).***

The usage of the particular frame type defines the quality and the compression ratio of the compressed video**.** I-frames **increase the quality** (and size), whereas the usage of B-frames **compresses better** but also produces **poorer quality**. The **distance** between two I-frames can be seen as a **measure for the quality** of an MPEG-video. In practice following sequence showed to give good results for quality and compression level: IBBPBBPBBPBBIBBP.

Introduction to Digital Audio

Some related definitions

Analog refers to something that is continuous.

Digital refers to something that discrete.

Analog signals : continuous waves that transmit information by altering the ***amplitude*** and ***frequency*** characteristic of the wave.

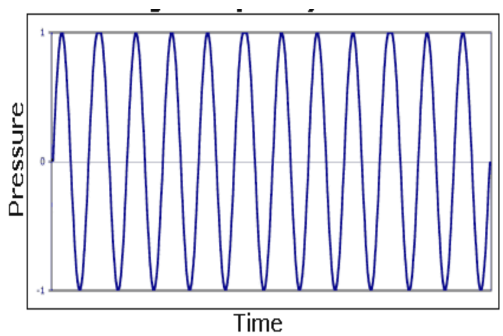
Digital signals : discrete pulses, either on or off, that convey information in a binary form that can be clearly interpreted by computer.

Analog Signals

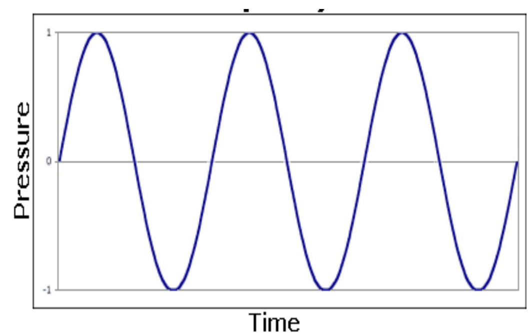
Frequency : the number of cycles per second.

Or the number of times a wave repeated during a specific time interval.

Units : Hertz (Hz).



High frequency



Low frequency

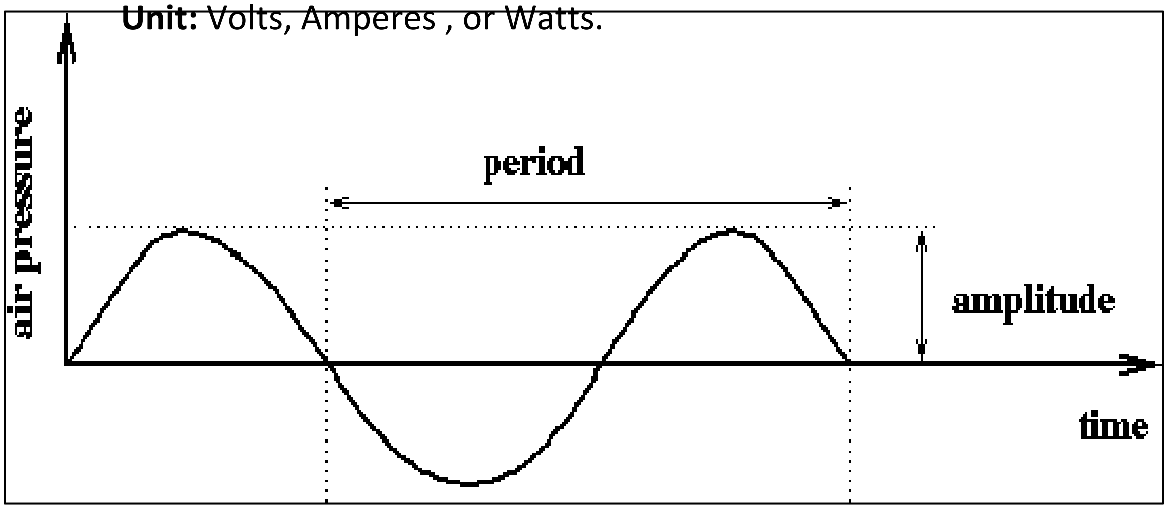
Period : refers to the amount of time, in second, a signal needs to ***complete*** one cycle.

$$Period = \frac{1}{\text{Frequency}}$$

Amplitude: The amplitude of wave is representing the **voltage levels**. It refers to half distance between the highest and lowest point in a wave, and it determine loudness of sound.

Amplitude measures how much **energy** is being transported by the wave.

Unit: Volts, Amperes , or Watts.



A sine wave

Bandwidth: the range of frequencies available in a communication channel, ***the greater the bandwidth the greater the channel capacity***.

To ***calculate the bandwidth*** subtracts the lowest frequency from the highest frequency of the range.

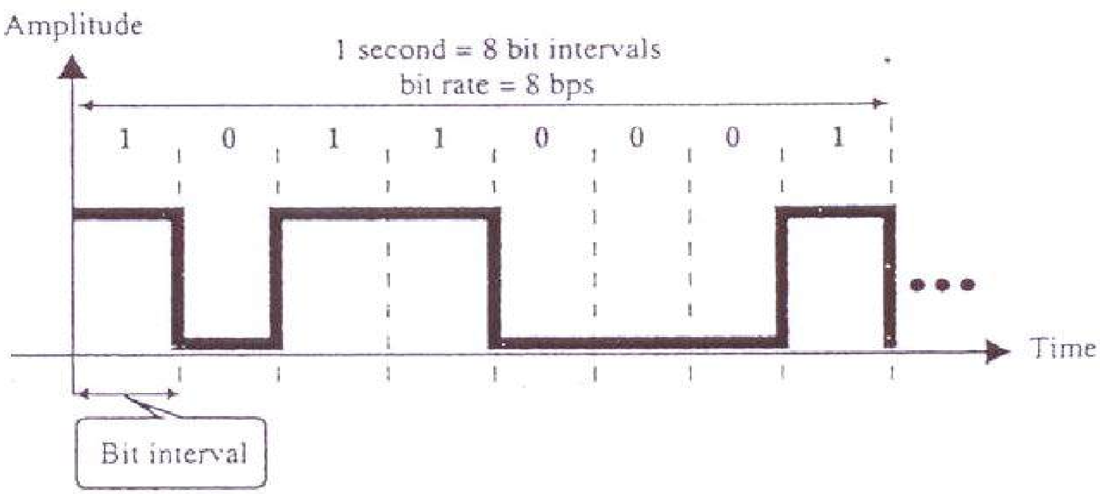
Digital Signals

Bit interval: is the time required to send one single bit.

Bit rate: is the number of bit intervals per second.

Or is the number of bits transmitted during one second.

Bit interval ≡ Period in analog; Bit rate ≡ Frequency in analog.



Bit rate and bit interval

Channel capacity: is the maximum bit rate a transmission medium can transfer.

1. Basic of Digital Audio

An audio signal are crucial for multimedia presentation and, in a sense, is the simplest type of multimedia data. However, some important differences between audio and image information cannot be ignored. For example, while it is customary and useful to occasionally ***drop a video frame*** from a video stream, to ***facilitate viewing speed***, we simply cannot do the same with sound information or all sense will be lost from that dimension.

1.1 What is Sound?

An intuitive definition: Sound is the sensation detected by our ears and interpreted by our brain in a certain way.

A scientific definition: Sound is a physical disturbance in a medium. It propagates in the medium as a pressure wave by the movement of atoms or molecules.

- Sound travels through all forms of matter: **gases, liquids and solid**. These are called the **medium**. Sound **cannot** travel through a **vacuum.**

Sound is a wave phenomenon like light, but it is macroscopic and involves molecules of air being compressed and expanded under the action of some physical device. For example, a speaker in an audio system vibrates back and forth and produces a longitudinal ***pressure wave*** that we perceive as sound.

Without air there is no sound, for example, in space, since sound is a pressure wave, it takes a continuous values, as opposed to digitized ones with a finite range. Nevertheless, if we wish to use a digital version of sound waves, we must form digitized representations of audio information.

Even though such pressure waves are longitudinal, they still have ***ordinary wave properties*** and behaviors, such as ***reflection, refraction*** (change of angle when entering a medium with a different density), and ***diffraction*** (being around an obstacle). This make the design of surround sound possible.

Since sound consists of measurable pressure at any 3D point, we can detect it by measuring the pressure level at a location, using a ***transducer to convert pressure to voltage levels.***

2. Digitization Audio

When sound is fed into a ***microphone***, an electronic analog signal is generated which represents the sound amplitude as a function of time. The signal is called an *analog audio signal.* An analog signal, such as audio, can be digitized to produce a digital signal. According to the Nyquist theorem, if the highest frequency of the signal is f, we need to sample the signal 2f times per second. There are other methods for digitizing an audio signal, but the principle is the same.

Figure (1) shows the one-dimensional nature of sound. Values change over time in amplitude: the pressure increases or decreases with time. The amplitude value is a continuous quantity. Since we are interested in working with such data in computer storage, we must digitize the analog signals (i.e., continuous-valued voltage) produced by microphones.

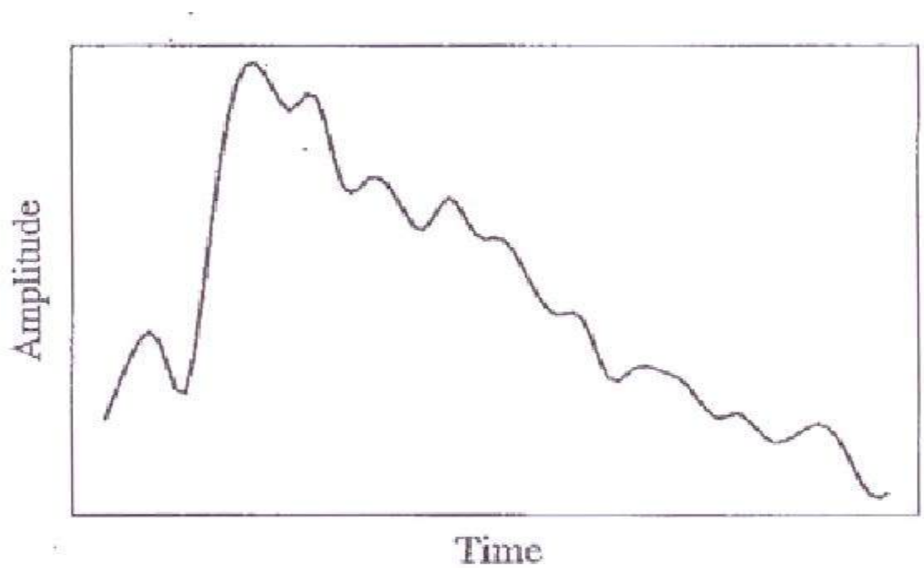


Figure (1) an analog signal : continuous measurement of pressure wave.

Digitization means **conversion of analog signals** to a stream of **numbers** – preferably integers for efficiency. Digitization required two processes; sampling and quantization.

Sampling: means measuring the amplitude of the signal at **equal intervals** (at evenly spaced intervals).

Quantization: means **assigning** interval **value** in a specific range to sampled instance.

Or, to restrict a variable quantity to discrete value rather that to continuous set of values.

There are two factors that determine fidelity of the original analog signal: the sampling rate and the bit depth or resolution of the sample. **Sampling Rate** is the number of samples that are used to represent one second of sound. The **Bit Depth** is the number of bits of information in each sample and it directly corresponds to the resolution of each sample. It may be 8,16,24 bit per samples.

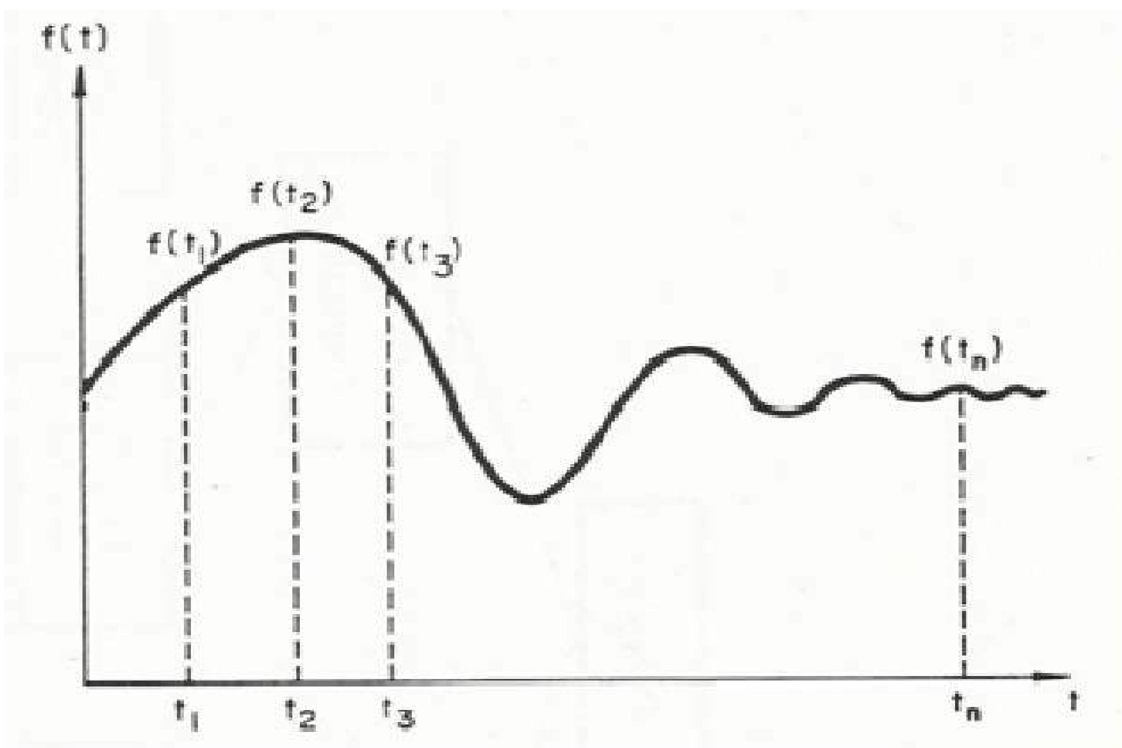


Figure (2) sampling of analog signal

For audio ***typical sampling rate*** are from **8 KHz** (8000 sample per second) to **48 KHz**. The ***human ear can hear from about 20 Hz*** (a very deep rumble) ***to as most as 20 KHz***; above this level, we enter the range of ultrasound. The human voice can reach approximately 4 KHz and we need to ***bound our sampling*** rate from below by at ***least double this frequency*** (according to Nyquist sampling rate, below). Thus we arrive at the useful range about 8 to 40 or so KHz.

While we have discussed only uniform sampling, with equally spaced sampling intervals, non uniform sampling is also possible.

To decide how to digitize audio data, we need to answer the following questions:

1. What is the sampling rate?
2. How finely is the data to be quantized, and is the quantization uniform?
3. How is the audio data formatted (i.e., what is the file format)?

3. Nyquist theorem

Nyquist theorem state that: the sampling rate must be ***at least*** two times the highest frequency.

Thus, for correct sampling we must use a sampling rate equal to at least twice the maximum frequency content in the signal. This is called the ***Nyquist rate.***

Ex

If the frequency spectrum of a signal has a bandwidth of 500 Hz with the highest frequency of 600 Hz, what should be the sampling rate according to Nyquist theorem?

Sol

According to Nyquist theorem, the sampling rate must be twice the highest frequency in the signal:

Sampling rate = 2 × (600) = 1200 samples/Second

4. Audio Quality versus Data Rate

The uncompressed data rate increases as more bits are used for quantization.

4.1 Data Rate

The term ***bandwidth***, in analog devices, refers to the range of frequencies available in a communication channel, the greater the bandwidth, the greater the channel capacity.

So for ***analog devices***, the ***bandwidth*** is expressed in frequency units, called Hertz (Hz), which is cycle per second. For ***digital devices,*** on the other hand, the amount of data that can be transmitted in a fixed bandwidth is usually expressed in the bits per second (bps) or bytes per amount of time. For either analog or digital, the term (***bandwidth***) expresses the ***amount of data*** that can be ***transmitted*** in ***fixed amount of time.***

Ex 1

What is the bandwidth of a signal that is ranges from 40 KHz to 4 MHz?

Sol

Bandwidth = Highest frequency – lowest frequency

Bandwidth = 4000 – 40 = 3960 KHz

= 3.96 MHz

Ex 2

If the bandwidth of a signal is 5 KHz and the lowest frequency is 52 KHz. What is the highest frequency?

Sol

Bandwidth = Highest frequency – lowest frequency

5 = highest frequency - 52

Highest frequency = 52+ 5 = 57 KHz.

Ex 3

What sampling rate is needed for a signal with a bandwidth of 10,000 Hz (1000 to 11,000 Hz) ? if the quantization (bit depth) is **eight bits** per sample , what is the bit rate?

Sol

According to Nyquist theorem, the sampling rate must be twice the highest frequency in the signal:

Sampling rate = 2 × 11,000 = 22,000 samples/Second

Each sample is quantized to eight bits:

Data rate = (22,000 samples/Second) × (8 bits/sample) = 176 Kbps

From the above, we can conclude that, Voice is sampled at 8000 samples per second with 8 bits per sample. This results in a digital signal of 64 kbps. Music is sampled at 44,100 samples per second with 16 bits per sample. This results in a digital signal of 705.6 kbps for monaural and 1.411 Mbps for stereo.

4.2 Signal-to-Noise Ratio (SNR)

In any analog system, random fluctuations produce noise added to signal, and the measured voltage is thus incorrect. The ratio of the power of the correct signal to the noise is called the signal-to-noise ratio (SNR). Therefore, the ***SNR is a measure of the quality of the signal.***

The SNR is usually measured in decibels (dB), where 1 dB is tenth of a bel. The SNR value, in units of dB, is defined in terms of base-10 logarithms of squared voltage:

$$SNR = 20 \log_{10} \frac{Vsignal}{Vnoise}$$

Using logarithms, we only have to deal with numbers in the range 0 through 11. In fact, this range is too small, and we typically multiply it by 10 or by 20, to get numbers between 0 and 110 or 220. This is the well-known (and sometimes confusing) **decibel system of measurement.**

Ex 1

If the signal voltage V_{signal} is 10 times the noise voltage V_{noise} , calculate the SNR.

Sol

SNR = 20 $\log_{10}(10)$ = 20 dB.

Ex2

If the noise voltage V_{noise} is 0.125 times the V_{signal}, calculate the SNR.

Sol

$$SNR = 20\log_{10}\frac{Vsignal}{Vnoise}$$

$$= 20\log_{10}\frac{1}{0.125} = 20\log_{10} 8 = 20 \times 0.903 = 18.06 \text{ dB}.$$

5. Compression of audio

Quantization and transformation of data are collectively known as coding of the data. ***Differences*** in signals between the present and previous time can effectively reduce the size of signal values and, most important, concentrate the histogram of pixel values (differences, now) into a much in signals between the present and previous time can effectively reduce the size of signal values and, most important, concentrate the histogram of pixel values (differences, now) into a much smaller range. The result of reducing the variance of values is that lossless compression methods that produce a bit stream with ***shorter bit*** lengths for ***more likely values***.

In general, producing quantized samples output for audio is called Pulse Code Modulation, or PCM. The differences version is called DPCM.

Finally, we may wish o compress the data, by assigning a bit stream that uses fewer bits for the most prevalent signal values.

Every compression scheme has three stages :

1. **Transformation** : the input data is transformed to a new representation that is easier or more efficient to compress. For example applying PCM to analog signals.
2. **Loss** : we may introduce loss of information. Quantization is the main lossy step. Here we use a limited number of reconstruction

levels, fewer than in the original signal. Therefore, quantization necessitates some loss of information.

3. **Coding** : here, we assign a codeword (thus forming a binary bit steam) to each output level or symbol. This could be a fixed-length code or a variable-length code, such as Huffman coding.

For audio signals, we first consider PCM, the digitization method. That enables us to consider lossless predictive coding as well as the DPCM scheme; these methods use ***differential coding***.