University of Technology

الجامعة التكنولوجية

Computer Science Department

قسم علوم الحاسوب

Multimedia 2

الوسائط المتعددة ٢

Prof. Dr. Abdulamir Abdullah

أ.د. عبد الأمير عبد الله

cs.uotechnology.edu.iq

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).

Output

Input

	Image	Description
Image	Image Processing	Computer Vision
Description	Computer Graphics	Other Data Processing

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 (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 that understand additional multimedia instructions.

– ***A consumer entertainment vendor***: interactive cable TV with hundreds of digital channels available, or a cable TV-like service delivered over a high-speed Internet connection.

– ***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 (**steganography**) which uses the available media as a cover to hide the secret data.

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

– Graphics, visualization, computer vision, data compression, networking, database systems – 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. “Augmented” reality: placing real-appearing computer graphics and video objects into the **real** scenes.

7. Including audio cues for where video-conference participants are located.
8. Building searchable features into new video, and enabling very high- to very low-bit-rate use of new, scalable multimedia products.
9. Making multimedia components *editable*.
10. Using voice-recognition to build an interactive environment, say a **web browser**.

1.3 Multimedia Research Topics and Projects

To the computer science researcher, multimedia consists of a wide variety of topics:

1. **Multimedia processing and coding**: multimedia content analysis, content-based multimedia retrieval, multimedia security, audio/image/video processing, compression, etc.
2. **Multimedia system support and networking**: network protocols, Internet, operating systems, servers and clients, quality of service (QoS), and databases.
3. **Multimedia tools, end-systems and applications**: hypermedia systems, user interfaces, authoring systems.
4. **Multi-modal interaction and integration**: "ubiquity" — web-everywhere devices, multimedia education including Computer Supported Collaborative Learning, and design and applications of virtual environments.

1.4 Current 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 ***graphics model***.
4. **3D capture technology**: allow **synthesis** of highly realistic facial animation from speech.
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**.

1.5 Multimedia and Hypermedia

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:

Hypermedia and Multimedia

- A **hypertext** system: We may think of a book as a linear medium, basically meant to be read from beginning to end, a hypertext meant to be read nonlinearly, by following links that point to other parts of the document, or to other documents (Fig. 1.1)
- **Hypermedia** : not constrained to be text-based, can include other media, e.g., graphics, images, and especially the continuous media — sound and video (e.g. home pages contain a link to video , images, and sound).

As we have seen, multimedia fundamentally means that computer information can be represented through audio, images, graphics, video, and animation in addition to traditional media (text and graphics). Hypermedia can be considered one particular multimedia application.

*** The World Wide Web (WWW)** — the best example of a hypermedia application.

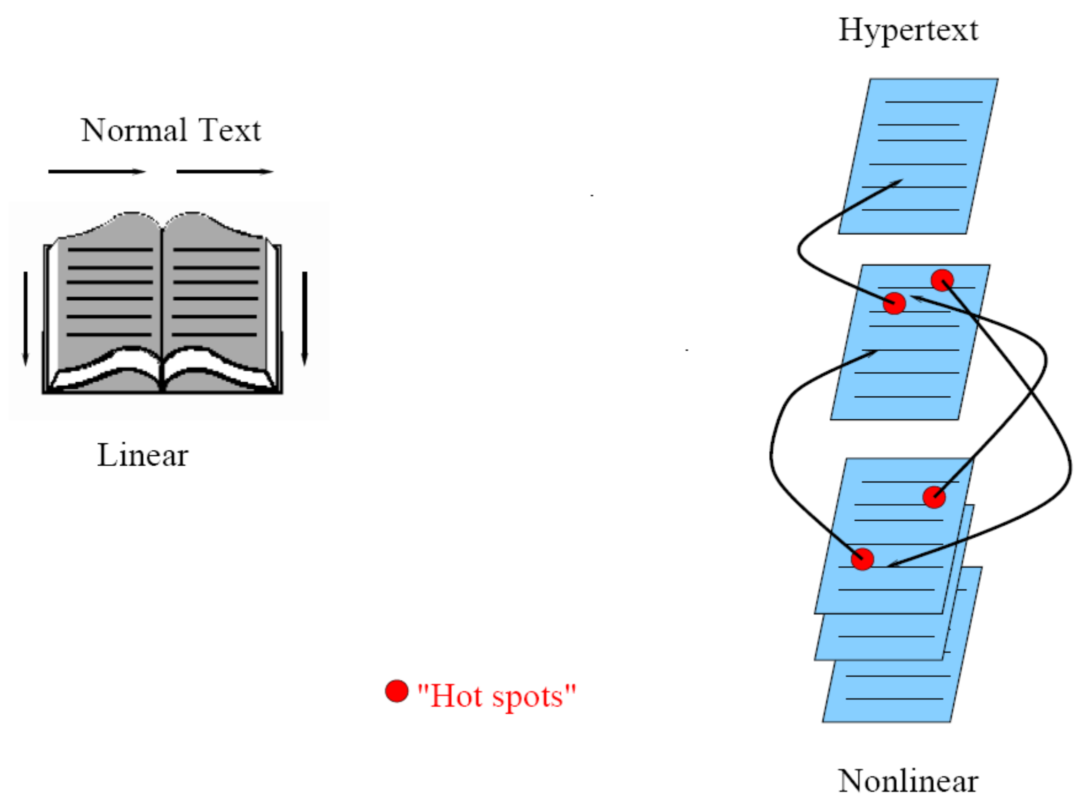


Figure 1.1 hypertext system

1.6. 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 TV-like 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 your self into virtual reality, you need ***software*** and special headgear, 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 this 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 an 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 users 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.

1.7 Internet

The internet , which is the largest computer network in the world, is actually, a network of networks. It is a collection of more than 200,000 individual computer networks owned by governments, universities, nonprofit groups, and companies. These interconnected networks exchange information seamlessly by using the ***same standards and protocols***. They are connected via high-speed, long-distance, backbone networks.

Thus, the internet forms a massive electronic communications network among businesses, consumers, government agencies, schools, and other organizations world-wide. Equally important, the internet has opened up exciting new possibilities that challenge traditional ways of interacting, communicating, and doing business. At the same time , the internet is raising new issues relating to culture and law, as its about business.

1.8 World Wide Web

Are the internet and the World Wide Web the same things? Many people believe that the Web is synonymous with the internet, but that is not the case. The internet functions as the ***transport mechanism***, and the World Wide Web(WWW) is the ***application*** that uses those transport functions. Other applications also run on the internet, with e-mail being the most widely used.

The World Wide Web is the biggest and the most widely known **information source** that is easily accessible and searchable. It consist of billion of interconnected documents (called web pages) which are authored by millions of people.WWW is an architecture framework for accessing linked documents spread out over million of machines all over the internet. The web would not be possible without the internet, which provide the communication network for the web to function.

The World Wide Web is a system with universally accepted ***standards for storing, retrieving, formatting, and displaying information*** via a client/server architecture. The Web handles all types of digital information , including text, hypermedia, graphics, and sound.

What makes the World Wide Web so graphically inviting and ***easily navigable*** is that this international collection of servers **(1)** contains information in multimedia form and **(2)** is connected by hypertext links.

1. Multimedia form _ what makes the Web graphically inviting:

   Whereas e-mail messages are generally text, the web provides information in multimedia form_ graphics, video, and audio as well as text. You can see color pictures, animation, and full-motion video. You can download music. You can listen to radio broadcasts. You can have telephone conversation with others.

2. Use of hypertext and hypermedia what makes the Web easily navigable:

   Whereas with e-mail you can connect only with specific addresses you know about, with the Web you have hypertext. **Hypertext** is a system in which documents scattered across many internet sites are directly linked, so that a word or phrase in one document becomes a connection to a document in a different places.

From the most commonly terms we encounter on almost daily are:

- **Web site_ the domain on the computer** : a website represent a centrally managed group of web pages , containing text , images and all types of multimedia files presented to the internet users in an easily accessible way.

  a computer with a domain name is called a ***site***. When you decide to buy books at the online site of bookseller, you would visit its Web site e.g. www. uotechnology.edu.iq ; the ***Web site*** is the location of a Web domain name in a computer somewhere on the internet.

- **Web pages_ the documents on a Web site** : a Web site is composed of a web page or collection of related Web pages. A ***Web page*** is a document on the World Wide Web that can include text , pictures, sound, and video. the first page you see at a Web site is like the title page of a book. This is the ***home page***, or welcome page, which identifies the Web site and contains links to other pages at the site.

Goals for World Wide Web

The world wide web is the largest and the most commonly used hypermedia application. It's popularity is due to the ***amount of information available from web servers, the capacity to post such information, and the ease of navigating such information with a web browser***.

The W3C has listed the following ***goals*** for the WWW:

1. Universal access of web resources (by everyone everywhere).
2. Effectiveness of navigating available information.
3. Responsible use of posted material.

1.9 Multimedia on the Web

Many Web sites are multimedia , using a combination of text, images, sound, video or animation.

- **Text and images**: you can call up all kinds of text documents on the Web, such as newspapers, magazines, famous speeches, and works of literatures. you can also view image such as scenery , famous paintings, and photographs. Most Web pages combine both text and images.
- **Animation** : animation is a rapid sequencing of still images to create the appearance of motion. Animation is also used in online video games .
- **Video** : video can be transmitted in ***two ways***. A file, such as a movie or video clip, may have to be completely **downloaded** before you can view it. This may take several minutes in some cases. A file may be displayed as a streaming video and viewed while it is still being downloaded to your computer. **Streaming video** is the process of transferring data in continuous flow so that you can begin viewing a file even before the end is sent.
- **Audio** : audio, such as sound or music files, may also be transmitted in two ways: ***(1) downloaded completely*** before they can be played or ***(2) downloaded as a streaming audio***, allowing you to listen to the file while the data is still being downloaded to your computer.

2.1 **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.

2.2 **computer graphics modes**

There are two extremely ***modes*** :

1. The passive (none interactive mode).
2. The interactive mode.

A typical system may be a hybrid of both modes.

1. The passive mode.

In this mode a system usually operate in a batch environment. The input is usually an ***already written program*** saved at a disk, the output devices are ***usually hard copy*** i.e. they provide a permanent pictures. Such as ***printers and plotters***.

2. The interactive mode

In this mode the user and the computer interact or converse on-line. The input device used with this mode are ***mouse, keyboard, joystick***, while the output device **must** be a ***display monitor***.

The main reason for the effectiveness of interactive computer graphics in many applications is the ***speed*** with which the user of the computer can assimilate the display information.

2.3 **How the Interactive Graphics display works**

The modern graphics display is extremely simple in construction. It consists of three components:

1- A digital memory, or frame buffer, in which the displayed image is stored as a ***matrix of intensity*** values.
2- A display monitor.
3- A display controller, which is a simple ***interface*** that passes the contents of the frame buffer (memory)to the monitor.

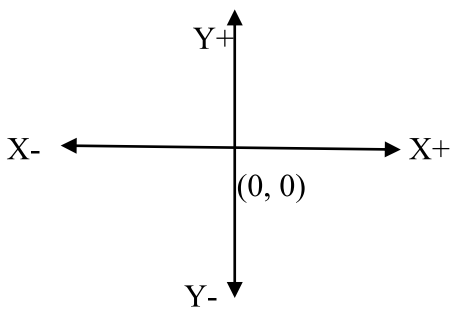
Inside the ***frame buffer*** the image is stored as a pattern of binary digital numbers, which represent a ***rectangular array*** of picture elements, or pixel.

The **pixel** is the smallest ***addressable*** screen element. In the simplest case where we wish to store only black and white images, we can represent black pixels by 0's in the frame buffer and white pixels by 1's. ***The display controller simply reads each successive byte of data from the frame buffer and convert each 0 and 1 to the corresponding video signal.*** This signal is then fed to the monitor. If we wish to change the displayed picture all we need to do is to change or modify the frame buffer contents to represent the new pattern of pixels.

2.4 Cartesian coordinate system

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

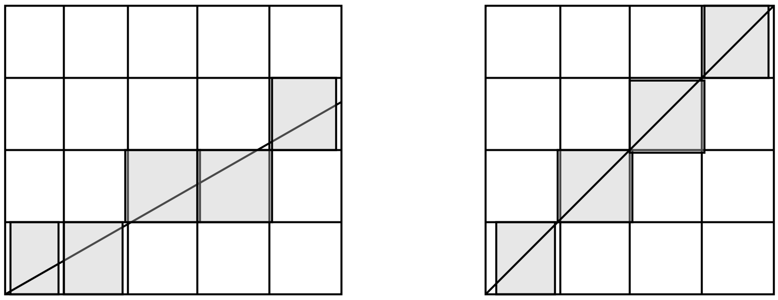
In a two-dimensional plane, we pick any point and single it out as a reference point called the origin. Through the origin we construct two perpendicular number lines called axes. These are labeled the X axis and the Y axis. Any point in two dimensions in this X-Y plane can be specified by a pair of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



2.5 Raster scan refresh graphics display (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.

The most common method of implementing a raster Display Monitor graphics device utilize a frame buffer. A frame buffer is a large, contiguous piece of computer memory. As a minimum there is one memory bit for each location or pixel in the raster. This amount of memory is called a bit plane.

A 512 × 512 element square raster requires 2^18 = 262144 memory bits in a single bit plane. The picture is built up in the frame buffer 1 bit at a time.

Since a memory bit has only two states (0 or 1), a single bit plane yields a black and white display. Color or gray levels can be incorporated into a frame buffer raster graphics device by using additional bit planes.

The ***capacity of the frame buffer*** depends on:

1. the number of bits representing each pixel.
2. the number of pixels per scan line.
3. the number of the scan lines.

The Display Monitor must be ***refreshed*** by repeatedly passing to it the image to be displayed. The image must be transmitted to the display point by point. Unless the entire image can be transmitted at least 25 times a second, it will begin to ***flicker***.

2.6 Pixels and Frame buffer

We can not represent an infinite number of points (pixels) on a computer, just as we can not do that with numbers. The machine is finite, and we are limited to a finite number of points making up each line.

The measure of the Resolution of the display device depends on :

1. The maximum number of points (pixels) which a line can have.

2. The number of lines in the image.

The greater the number of points, the higher the resolution.

Resolution is the number of visibly distinct dots (Pixels) that can be displayed in a given area of the screen.

2.7 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.

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:= y1 – x1 * 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;
```

2.8 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 :

$$ABS\ (X_2\text{-}X_1)$$

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

$$ABS\ (Y_2\text{-}Y_1)$$

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

The Length estimates is 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

Features of DDA

1- The algorithm is orientation dependent
2- The end point accuracy deteriorates
3- The algorithm suffer from the fact that it must be performed using floating point arithmetic

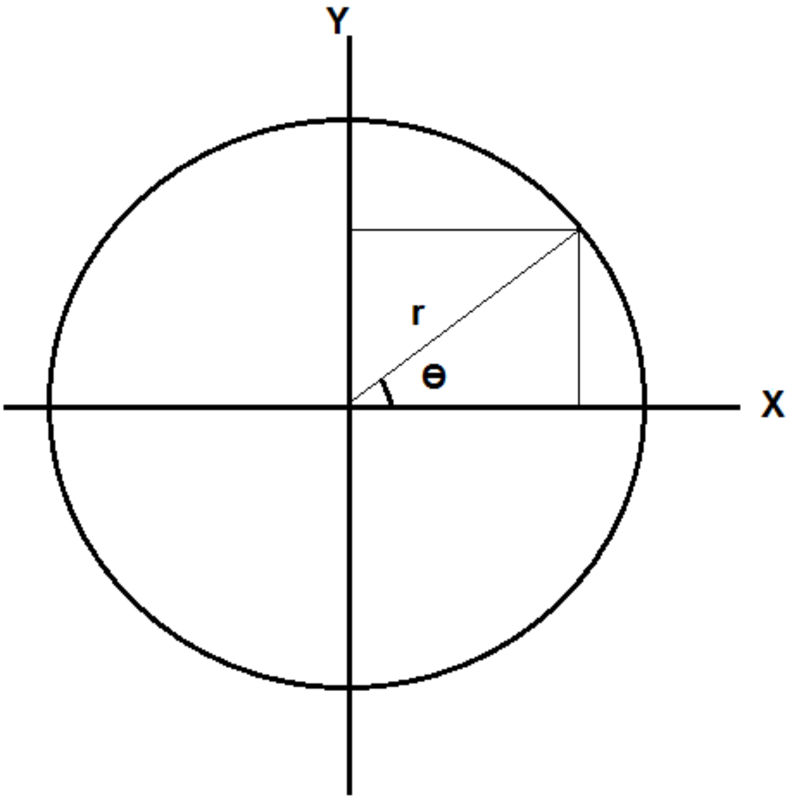
2.9 **Circle Drawing algorithms**

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 ;
```

2.10 **Bresenham algorithm for circle drawing**

To begin with circle drawing, note that only one octant of the circle need to be generated. The other parts can be obtained by successive reflections. To drive Bresenham circle generation algorithm , consider the first quadrant of the origin centered circle.

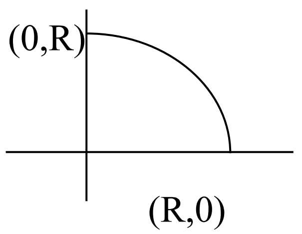
Notice that, if the algorithm ***begins at X=0 and Y=R*** (R is the radius) , then for clockwise generation of the circle, Y must be decreasing while X is increasing in the first quadrant.

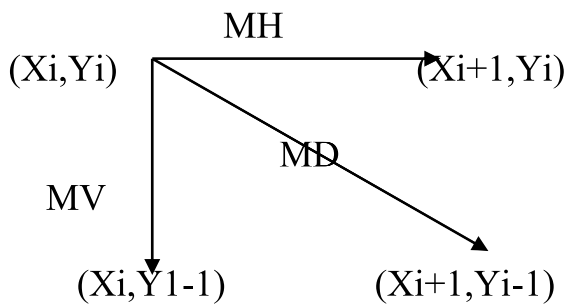
Similarly if the algorithm ***begins at Y=0 and X=R*** then for counterclockwise generation of the circle, X must be decreasing while Y is increasing in the first quadrant.

For clockwise generation of the circle there are only three possible selections for the next pixel which best represent the circle:

1- Horizontal to the right
2- Diagonally downward to the right
3- Vertically downward

These are labeled MH , MD , MV





The algorithm choose the pixel (the movement)which minimize the square of the distance between one of these pixels and the true circle.

The distance in the three cases are measured as :

Case 1 : MH = | (Xi+1)^2 + (Yi)^2 - R^2 |
Case 2 : MD = | (Xi+1)^2 + (Yi-1)^2 - R^2 |
Case 3 : MV = | (Xi)^2 + (Yi-1)^2 - R^2 |

The difference between the square of the distance from the center of the circle to the diagonal pixel at (Xi+1 , Yi-1) and the distance to a point on the circle R^2 is :

Di= (Xi+1)^2 + (Yi-1)^2 – R^2

1- If Di<0 then the diagonal point (Xi+1,Yi-1) is ***inside*** the actual circle i.e. we use case 1 or case 2 .

It is clear that either the pixel at (Xi+1,Yi) == MH or that the pixel at (Xi+1,Yi-1) == MD must be chosen.

ə = | (Xi+1)^2 + (Yi)^2 – R^2| - | (Xi+1)^2 + (Yi-1)^2 – R^2|

if ə <0 then the difference from the actual circle to the diagonal pixel (MD) is greater than that to the horizontal pixel (MH).

If ə <0 choose MH (Xi+1,Yi)
If ə >0 choose MD (Xi+1, Yi-1)
The horizontal move has been selected when ə=0 ; i.e. when the distance are equal.

2- if Di>0 then the diagonal point (Xi+1,Yi-1) is ***outside*** the actual circle i.e. we use case 2 or case 3 .

β = | (Xi+1)^2 + (Yi-1)^2 – R^2 | - | (Xi)^2 + (Yi-1)^2 + R^2 |

if β <=0 choose MD (Xi+1, Yi-1)
if β >0 choose MV (Xi , Yi-1)

3- if Di=0 then we choose the pixel at (Xi+1 , Yi-1) i.e MD.

In Summery :

1. Di < 0
   ə ≤ 0 , then choose pixel at (Xi+1 , Yi) i.e. MH
   ə > 0 , then choose pixel at (Xi+1 , Yi-1) i.e. MD
2. Di > 0
   β ≤ 0 , then choose pixel at (Xi+1 , Yi-1) i.e. MD
   β > 0 , then choose pixel at (Xi , Yi-1) i.e. MV
3. Di = 0
   choose pixel at (Xi+1 , Yi-1) i.e. MD

Algorithm 4 : *Bresenham circle algorithm (for the first quadrant)*

```
    Xi=0
    Yi=R
    Di= 2 (1-R)
    Limit = 0

 1:  PutPixel (Xi , Yi)
    If Yi ≤ Limit then goto 4
    If Di < 0 then goto 2
    If Di > 0 then goto 3
    If Di = 0 then goto 20

2:  ə = 2Di + 2Yi -1
    If ə ≤ 0 then goto 10
    If ə > 0 then goto 20

3:  β= 2Di – 2Xi -1
    If β ≤ 0 then goto 20
    If β > 0 then goto 30

10: Xi = Xi + 1              { MH }
    Di = Di + 2Xi + 1
    Goto 1

20: Xi = Xi +1         { MD }
    Yi = Yi - 1
    Di = Di + 2Xi – 2Yi + 2
    Goto 1

30: Yi = Yi -1    {  MV }
     Di = Di – 2Yi + 1
     Goto 1

4:  Finish
```

Example 5 :

Draw a circle with R=4

Plot (X,Y)	Di	∂	β	Xi	Yi
	-6	-	-	0	4
(0,4)	-3	-5	-	1	4
(1,4)	-3	1	-	2	3
(2,3)	4	-1	-	3	3
(3,3)	1	-	1	3	2
(3,2)	9	-	-5	4	1
(4,1)	10	-	9	4	0
(4,0)	-3				

2.11 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.

There are two complementary points of view for describing object movement:

1- Geometric transformation: the object itself is moved relative to a stationary coordinate system or background.
Geometric transformation is applied to each point of the object.

2- Coordinate transformation: the object is held stationary while the coordinate system is moved relative to the object, for example the motion of a car in a scene, we can keep the car fixed while moving the background scenery.

Geometric transformation

Display objects are defined by sets of coordinates points.

Geometric transformation are procedures for calculating new coordinate positions for these points , as required by a specified change in size and orientation for the object.

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 orginal 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

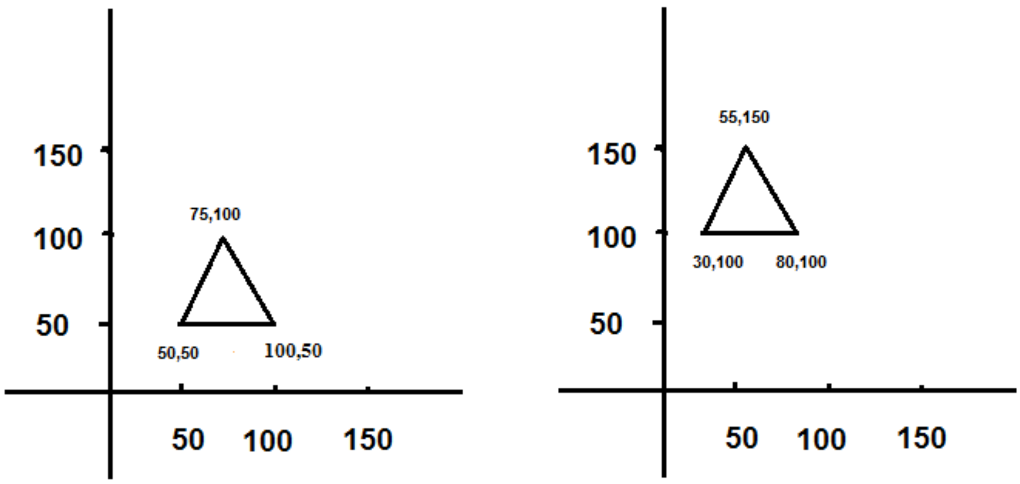
$$X'_2= 9 + 3= 12$$
$$Y'_2= -6 + 2= - 4$$

By using the matrix representation

$$\begin{bmatrix} X'_1 & Y'_1 & 1 \\ X'_2 & Y'_2 & 1 \end{bmatrix} = \begin{bmatrix} -4 & 3 & 1 \\ 9 & -6 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

Example 2

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



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

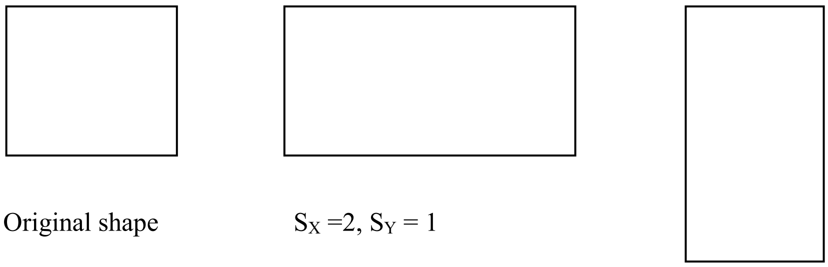
S_X is the scale factor in the X direction
S_Y is the scale factor in the Y direction

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 $\quad$ $S_X=2, S_Y=1$ $\quad$ $S_X=1, S_Y=2$

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)

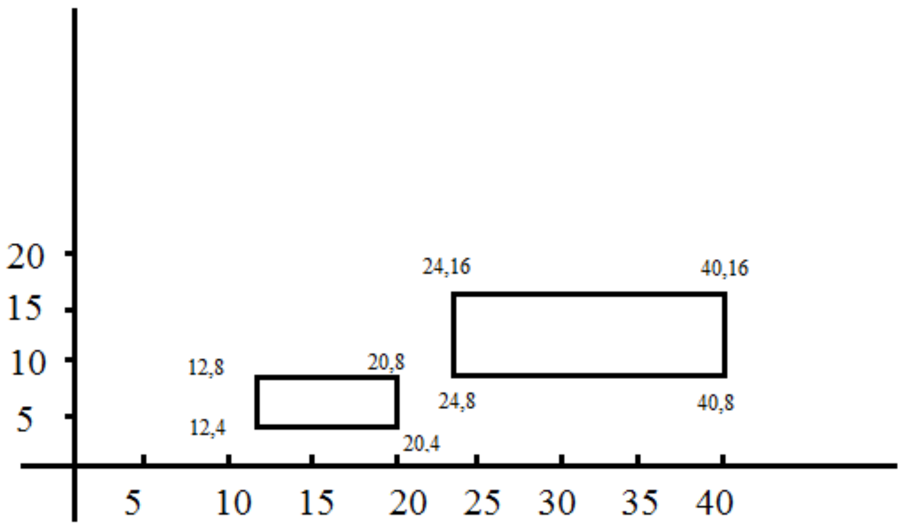
$X'_4 = 20 * S_X = 20 * 2 = 40$

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

Solution : (By using matrices)

$$\begin{bmatrix} X'_1 & Y'_1 & 1 \\ X'_2 & Y'_2 & 1 \\ X'_3 & Y'_3 & 1 \\ X'_4 & Y'_4 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 4 & 1 \\ 20 & 4 & 1 \\ 12 & 8 & 1 \\ 20 & 8 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 24 & 8 & 1 \\ 40 & 8 & 1 \\ 24 & 16 & 1 \\ 40 & 16 & 1 \end{bmatrix}$$



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

1: نقل الشكل الى نقطة الأصل
2 : تكبير الشكل
3 : ارجاع الشكل الى مكانة الأصلي

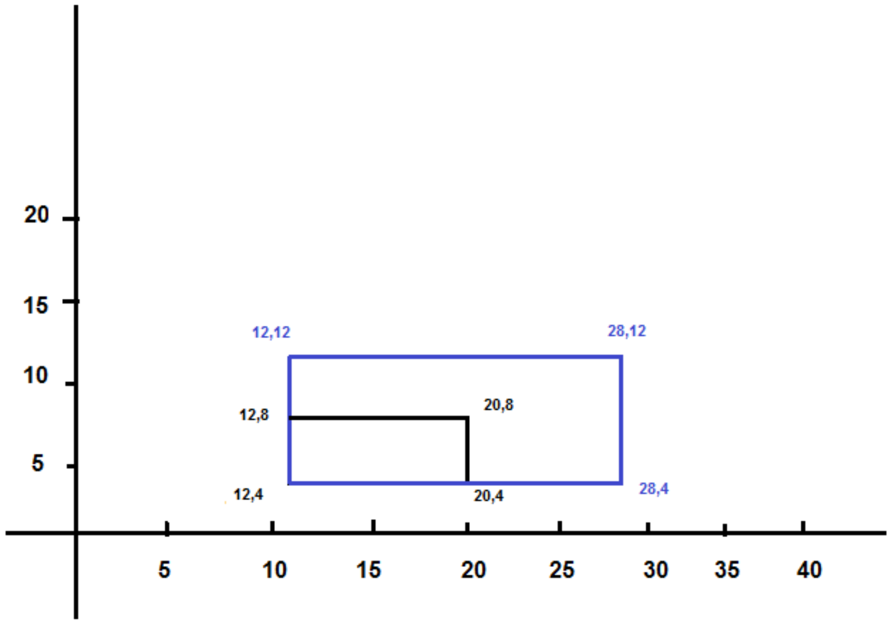
Directly using the equation , we get :

$X' = X_f + (X - X_f) S_X$;
$X'_1 = 12 + (12 – 12) 2 = 12$
$X'_2 = 12 + (20 – 12) 2 = 28$
$X'_3 = 12 + (12 – 12) 2 = 12$
$X'_4 = 12 + (20 –12) 2 = 28$

$Y' = Y_f + (Y - Y_f) S_Y$;
$Y'_1 = 4 + (4 – 4) 2 = 4$
$Y'_2 = 4 + (4 – 4) 2 = 4$
$Y'_3 = 4 + (8 – 4) 2 = 12$
$Y'_4 = 4 + (8 – 4) 2 = 12$

So the new coordinates will be :

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

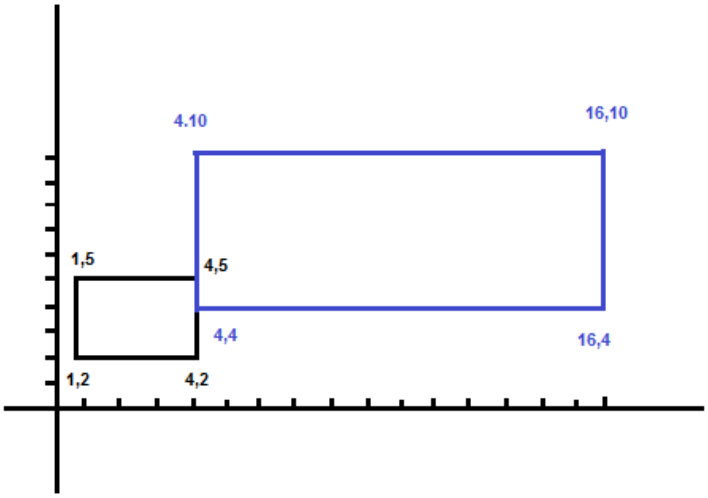


<u>Example 3</u>: Scale the square (1,2) , (4,2) , (1,5) , (4,5) with 4 units in the X-axis ,and 2 units in the Y-axis

Solution:

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

$$= \begin{bmatrix} 4 & 4 & 1 \\ 16 & 4 & 1 \\ 4 & 10 & 1 \\ 16 & 10 & 1 \end{bmatrix}$$

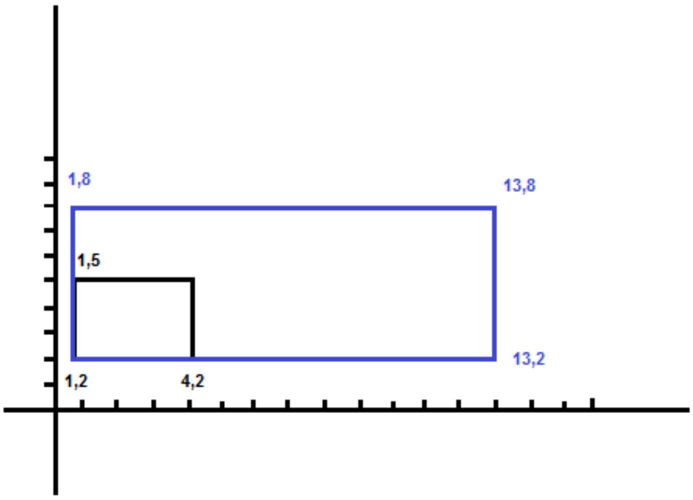


Its obvious that the scale operation above cause a change in the rectangular position. So to avoid this drawback we have to apply the three steps of scaling , Directly using the last equations, we get :

$X' = X_f + (X - X_f)\, S_X$;
$X'_1 = 1 + (1 - 1)\, 4 = 1$
$X'_2 = 1 + (4 - 1)\, 4 = 13$
$X'_3 = 1 + (1 - 1)\, 4 = 1$
$X'_4 = 1 + (4 - 1)\, 4 = 13$

$Y' = Y_f + (Y - Y_f)\, S_Y$;
$Y'_1 = 2 + (2 - 2)\, 2 = 2$
$Y'_2 = 2 + (2 - 2)\, 2 = 2$
$Y'_3 = 2 + (5 - 2)\, 2 = 8$
$Y'_4 = 2 + (5 - 2)\, 2 = 8$

So the new coordinates will be :
(1,2) , (13,2), (1,8), (13,8)



3- Rotation

Is the transformation of the object along a circular path.

We specify this type of transformation by a rotation angle Θ , which determine the amount of the rotation to be performed for each vertex of the polygon

$$\left.\begin{aligned} X &= r\cos\phi \\ Y &= r\sin\phi \end{aligned}\right\}\quad \text{from the circle equation}$$

$$X' = r\cos(\Theta + \phi)$$
$$Y' = r\sin(\Theta + \phi)$$

From geometric we have the below rules :

$\cos(a + b) = \cos a \cos b - \sin a \sin b$

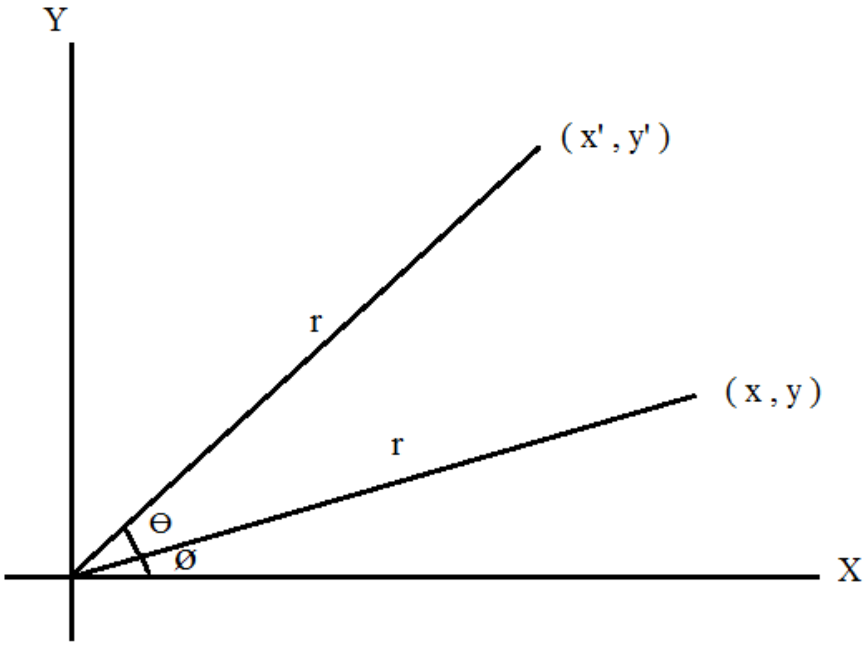
$\sin(a + b) = \sin a \cos b + \sin b \cos a$

hence :

$$X' = r\cos\Theta\cos\phi - r\sin\Theta\sin\phi$$
$$Y' = r\sin\Theta\cos\phi + r\cos\Theta\sin\phi$$

$X' = X\cos\Theta - Y\sin\Theta$ where $X = r\cos\phi$, $Y = r\sin\phi$

$Y' = X\sin\Theta + Y\cos\Theta$



Note that the direction of rotation is counterclockwise if Θ is a positive angle and clockwise if Θ is a negative angle.

Rotation about the origin

The rotation matrix which is used to rotate an object about the origin in anticlockwise direction is :

$$[\,X' \quad Y' \quad 1\,] = [\,X \quad Y \quad 1\,] \begin{bmatrix} \cos\Theta & \sin\Theta & 0 \\ -\sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Or, in equation :

$$X' = X * \text{Cos}\,\Theta - Y * \text{Sin}\,\Theta$$
$$Y' = X * \text{Sin}\,\Theta + Y * \text{Cos}\,\Theta$$

When $\Theta=90$, the matrix that cause a rotation through an angle of 90 ($\Pi/2$) is

0	1	0
-1	0	0
0	0	1

When $\Theta=180$

-1	0	0
0	-1	0
0	0	1

When $\Theta=270$

0	-1	0
1	0	0
0	0	1

When $\Theta=360$

1	0	0
0	1	0
0	0	1

Example 1: rotate the line $P_1(1,4)$ and $P_2(3,1)$ anticlockwise 90 degree.
Solution:

$$\begin{bmatrix} X'_1 & Y'_1 & 1 \\ X'_2 & Y'_2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 \\ 3 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -4 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

Rotate about a specific point (X_f,Y_f)

We have to follow three steps:

First: translate the points (and the object) so that the point (X_f,Y_f) lies on the origin

$X_1= X - X_f$
$Y_1= Y - Y_f$

Second: rotate the translated point (and the translated object) by Θ degree about the origin to obtain the new point (X_2,Y_2)

$X_2 = X_1 * \cos \Theta - Y_1 * \sin \Theta$
$Y_2 = Y_1 * \cos \Theta + X_1 * \sin \Theta$

Third : Back translation

$X_3= X_2 + X_f$
$Y_3= Y_2 + Y_f$

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

$X' = X_f + (X - X_f) \cos \Theta - (Y - Y_f) \sin \Theta$
$Y' = Y_f + (Y - Y_f) \cos \Theta + (X - X_f) \sin \Theta$
3 1 2 1 2

1 : نقل الشكل الى نقطة الأصل
2 : تدوير الشكل
3 : ارجاع الشكل الى مكانة الأصلي

Note:: Rotation in clockwise direction :

In order to rotate in clockwise direction we use a negative angle, and because :

$\cos(-\Theta)= \cos \Theta$
$\sin(-\Theta)= - \sin \Theta$

So the matrix will be (***for clockwise rotation***):

$$[\; X' \quad Y' \quad 1 \;] = [\; X \quad Y \quad 1 \;] \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Or, in equation (***for clockwise rotation***):

$$X' = X * \text{Cos}\,\Theta + Y * \text{Sin}\,\Theta$$
$$Y' = -X * \text{Sin}\,\Theta + Y * \text{Cos}\,\Theta$$

Example 2: Rotate the square (2,1),(4,1),(2,3),(4,3) counterclockwise with $\Theta = 45$ around the point (2,1)

Solution : First: translate the square by T_X=-2 and T_Y=-1

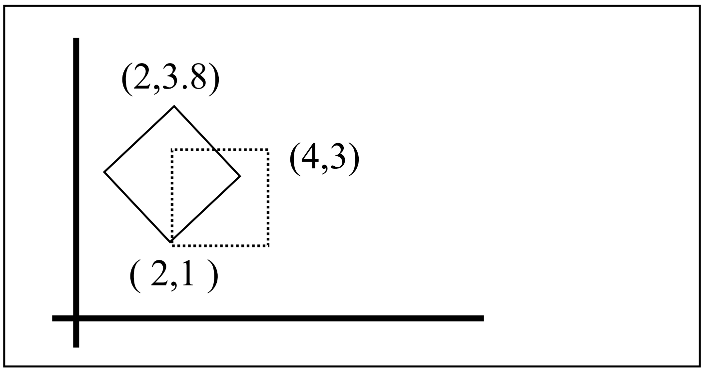
(2,1) ===> (0,0)
(4,1) ===> (2,0)
(2,3) ===> (0,2)
(4,3) ===> (2,2)

Second: Rotate by Θ=45

(0,0) ===> (0 , 0)
(2,0) ===> (1.414 , 1.414)
(0,2) ===> (-1.414 , 1.414)
(2,2) ===> (0 , 2.828)

Third: Back translation by T_X= 2 and T_Y= 1

(0 , 0) ===> (2 , 1)
(1.414 , 1.414) ===> (3.414 , 2.414)
(-1.414 , 1.414) ===> (0.586 , 2.414)
(0 , 2.828) ===> (2 , 3.828)



4-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.

4-1:Reflection on the X axis

1	0	0
0	-1	0
0	0	1

OR Xnew= X
Ynew= -Y

4-2:Reflection on the Y axis

-1	0	0
0	1	0
0	0	1

OR Xnew= - X
Ynew= Y

4-3:Reflection on the origin

-1	0	0
0	-1	0
0	0	1

OR Xnew= - X
Ynew= -Y

4-4:Reflection on the line Y=X

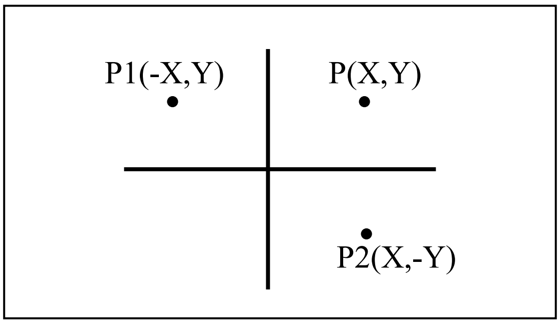
0	1	0
1	0	0
0	0	1

OR Xnew= Y
Ynew= X

4-5: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

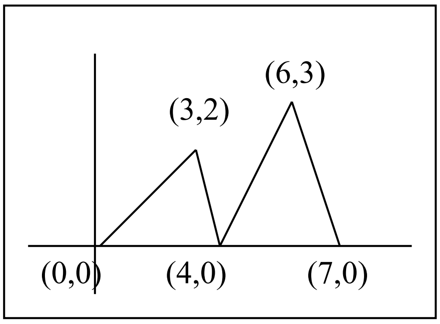
*

0	1	0
1	0	0
0	0	1

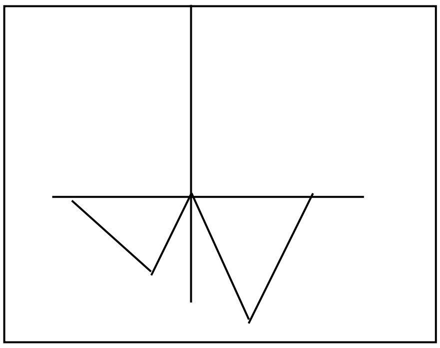
=

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

5-Shearing

It produces a distortion of an object. There are two types of shearing:-

1- X shearing
It transform the point (X,Y) to the point (Xnew,Ynew) where

Xnew= X+Shx * Y   where Shy≠0
Ynew= Y

The matrix is

1	0	0
Shx	1	0
0	0	1

2- Y shearing
It transform the point (X,Y) to the point (Xnew,Ynew) where

Xnew= X
Ynew= Y+Shy * X where Shy≠0

The matrix is

1	Shy	0
0	1	0
0	0	1

Y shearing moves a vertical line up or down depending on the sign of the shear factor Shy. A horizontal line is distorted into a line with slop Shy. And vis versa.

Example : Share the object (1,1) , (3,1) , (1,3) , (3,3) with
a: Shx=2
b: Shy=2

Solution :
a. Shx=2

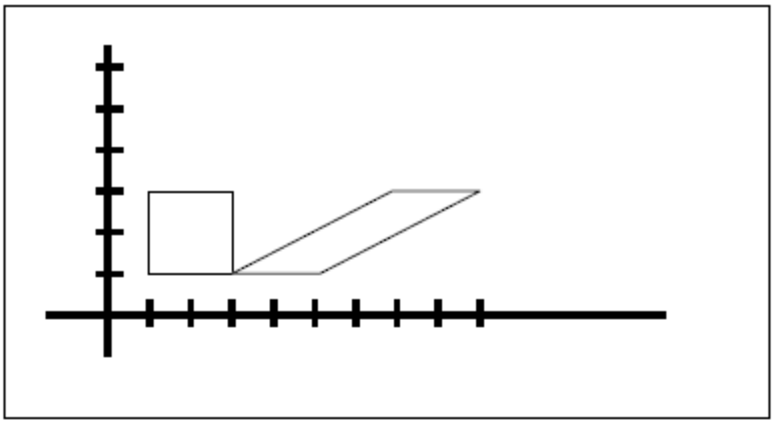
1	1	1
3	1	1
1	3	1
3	3	1

*

1	0	0
2	1	0
0	0	1

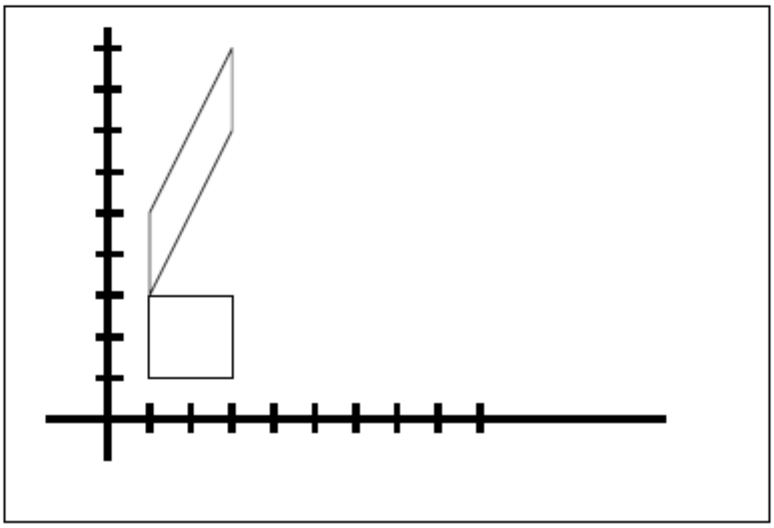
=

3	1	1
5	1	1
7	3	1
9	3	1



b- Shy

$$\begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 1 \\ 1 & 3 & 1 \\ 3 & 3 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 1 \\ 3 & 7 & 1 \\ 1 & 5 & 1 \\ 3 & 9 & 1 \end{bmatrix}$$



Additional examples

1. Translate the triangle (10,6) , (20,6) , (15,16) four units down and five units to the left .

2. Scale the rectangle (2, 2) , (8 , 2) , (2 , 6) , (8 , 6) so that it's new area will be **four times** as its original area and the point (8,6) be the fixed point.

3. Rotate the triangle (10,4) , (18,4) , (14,12) **270 degree** anticlockwise so that its center be the fixed point.

University of Technology

Computer Science Department

3rd Grade

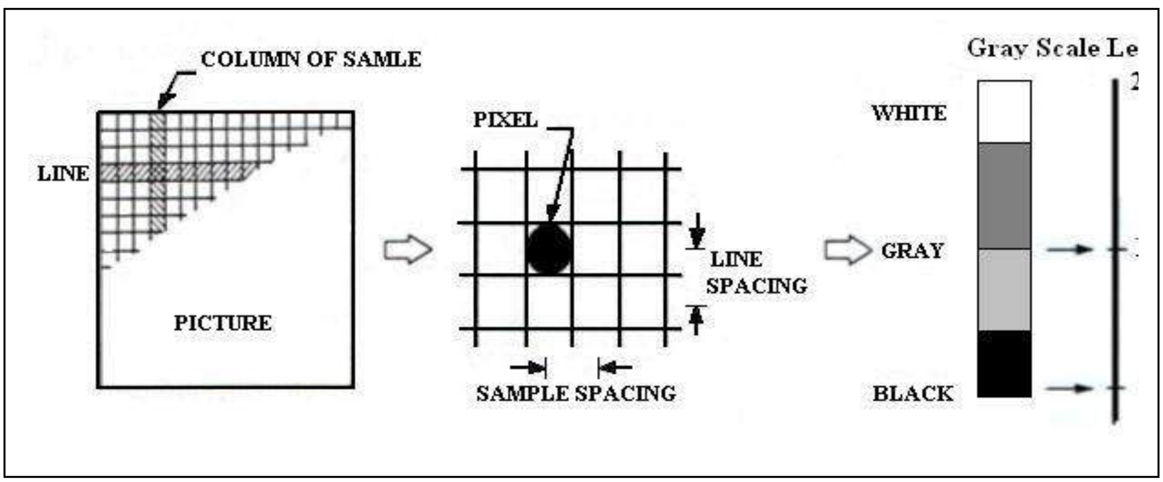
Multi Media

أستاذ المادة: أ.م. د. عبد الأمير عبد الله كريم

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 brightness is sampled and 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 integer. 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 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 a mount of useful information that can be extracted from an image .

1.2. Quantization

It is usual to digitize the value of the image function *f(x,y)* in addition to its spatial coordinates. This process of quantization involves replacing a continuously varying *f(x,y)* with a discrete set of quantization levels . 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 in the form of analog electrical signals and have seen how these are sampled to generate the digital image I (r, c).

The digital image I(r, c) 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 I(r, c) 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 I(r,c) 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.
2. In optical character recognition (OCR).
3. In text images.

Advantage of binary images

1. Reduce storage space (memory).
2. decrease processing 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

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.

2.3.1 RGB Color Model

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-a) we see a representation of a typical RGB color image. Figure (2-a) 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).

For many applications, RGB color information is transformed into a mathematical space that ***decouples*** the brightness information from the color information. After this is done, the image information consists of a one-dimensional brightness, or luminance, space and a two-dimensional color space. Now the two-dimensional color space does not contain any brightness information; but it typically contains information regarding the relative amounts of the different colors. An additional benefit of modeling the color information in this manner is that it creates a more people-oriented way of describing the colors.

2.3.2 HSL Color Model

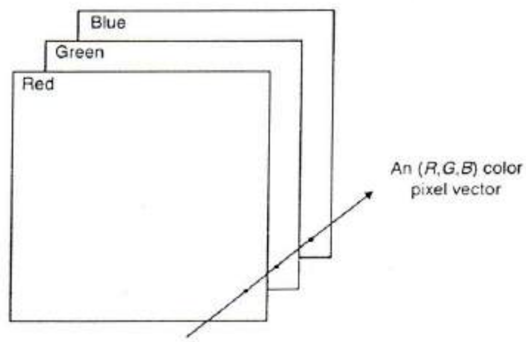
The **H**ue/**S**aturation/**L**ightness (HSL color transform allows us to describe colors in terms that became more readily understand (see Figure (2-b). The ***lightness*** is the brightness of the color, and the ***hue*** is what we normally think of as "color" (for example green, blue, or orange). The ***saturation*** is a measure of how much white is in the color (for example, pink is red with more white, so it is less saturated than a pure red). Most people can relate to this method of describing color. For example,

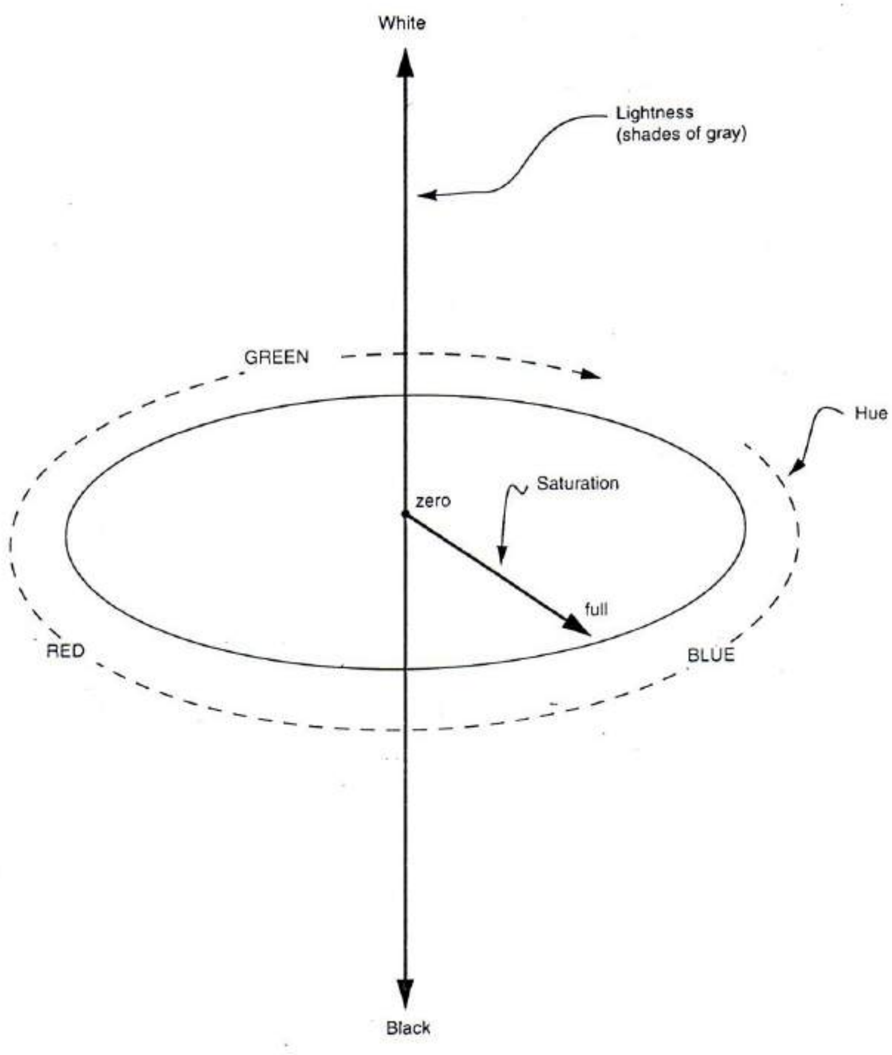
"a deep, bright orange" would a have a large intensity ("bright"), a hue of "orange" and a high value of saturation ("deep"). We can picture this color in our minds, but if we defined this color in terms of its RGB component R=245, G=110, and B = 20, most people would have no idea how this color appears. Because the HSL color space was developed based on heuristics relating to human perception, various methods are available to transform RGB pixel values into the HSL color space. Most of these are algorithmic in nature and are geometric approximations to mapping the RGB color cube into some HSL-type color space.

$$H = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}} \quad \ldots\ldots (1)$$

$$S = 1 - \frac{3 * \min(R,G,B)}{R+G+B} \quad \ldots\ldots (2)$$

$$L = \frac{R+G+B}{3} \quad \ldots\ldots (3)$$


Blue
Green
Red
An (R,G,B) color pixel vector


White
Lightness (shades of gray)
GREEN
Hue
Saturation
zero
full
RED
BLUE
Black

Figure(2) a. RGB image b.HSI image

2.3.3 YIQ Color Model

In this model the transmitted color signals are formed by encoding RGB image information into two parts, the *luminance* Y (brightness information) and the *chrominance* (color information) . the color *hue* information is given by the chrominance (phase) angle Q and the *saturation* information is given by the chrominance amplitude I.

The transformation of the RGB signals to the YIQ color space is given in the below equation :

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & -0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

This equation is a simple representation of the **NTSC** encoding scheme used in the **American** broadcast television.

2.3.4 YUV Color Model

Similar to YIQ representation, but for **European** TV transmission system, initially YUV coding was used for **PAL** and **SECAM** analogue video. The transformation of RGB signals to YUV color space is given by the below equation.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2.3.5 8-bits color images

GIF (Graphics Interchange Format) file format is commonly used in the World Wide Web. GIF files are limited to a maximum of 8 bit\pixel and allow for a type of compression called LZW (Lepel-Ziv-Welch). The 8 bit\pixel limitation does not mean that it does not support color images, it simply means that no more than 256 color (2^8) are allowed in an image. This is typically ***implemented*** by mean of lookup table (LUT), where the 256 colors are stored in a table, and 1 byte (8 bits) is used as an index (address) into that table for each pixel.

GIF images are indexed images where the colors used in the image are stored in a **Palette** **,**sometimes referred to as a color look-up-table. Each pixels is represented as a single byte, and the pixel data is an index to the color palette. The color of the palette are typically **ordered** from the most used color to the least used colors to **reduce** look-up time.

Color Look-Up Tables (LUTs)

- Store only the index of the color LUT for each pixel
- Look up the table to find the color (RGB) for that index
- LUT needs to be built when converting 24-bit color images to 8-bit: grouping similar colors (each group assigned a color entry)
- Possible for palette animation by changing the color map

/

8-bit index	Red	Green	Blue
0	R_0	G_0	B_0
1	R_1	G_1	B_1
2	R_2	G_2	B_2
:	:	:	:
:	:	:	:
254	R_{254}	G_{254}	B_{254}
255	R_{255}	G_{255}	B_{255}

color LUT for 8-bit Color Images

2.4 Multispectral images

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 3). 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.

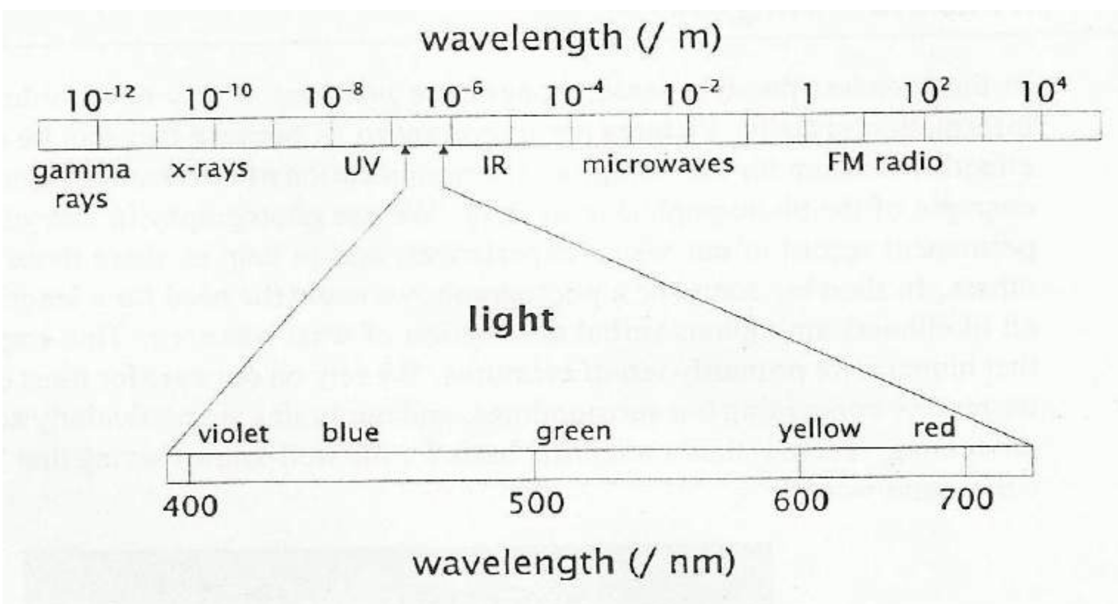
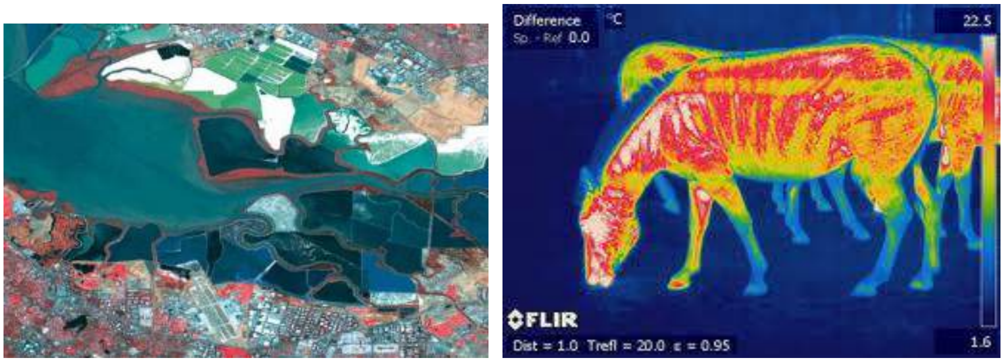


Figure (3) the spectrum of electromagnetic radiation



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.

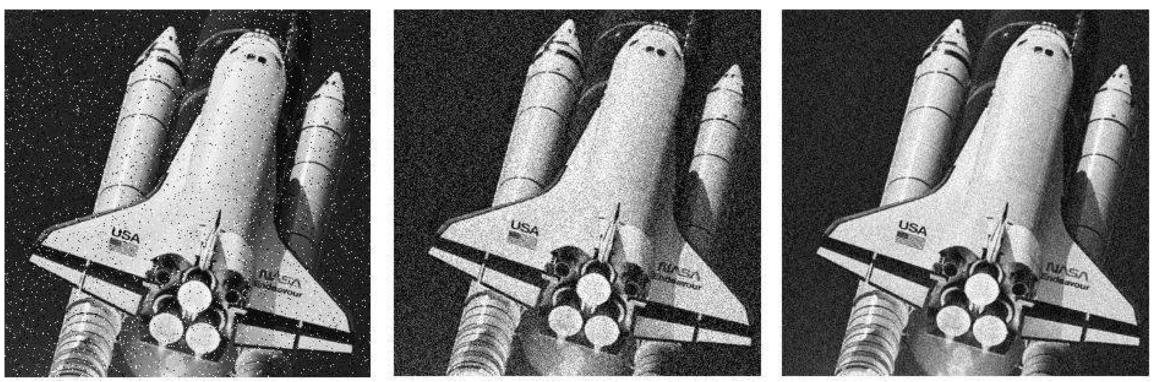
Addition can also be used to ***combine the information of two images***, such as an image morphing, in motion pictures.

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;
```



(a) (b) (c)

Figure (4) a) noisy image b) average of five observation c) average of ten observation

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

write out-array to out-image;
```

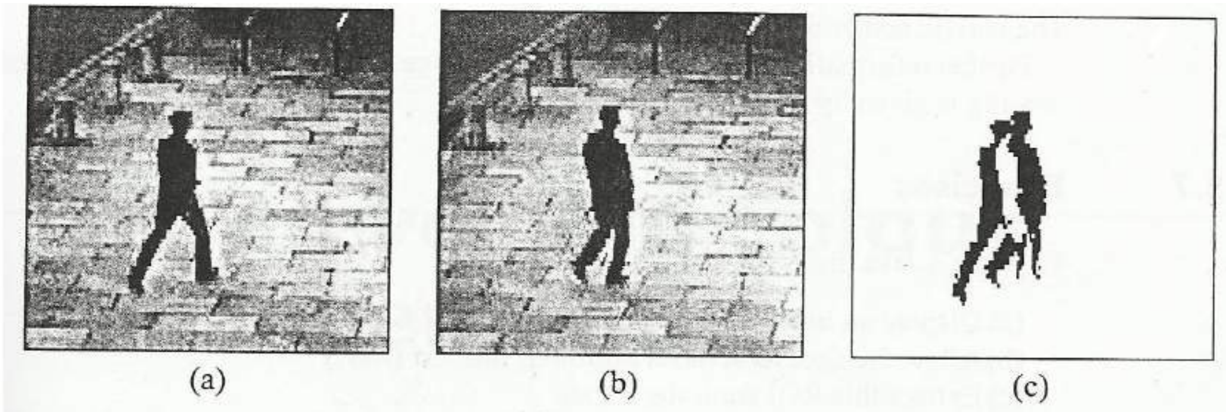
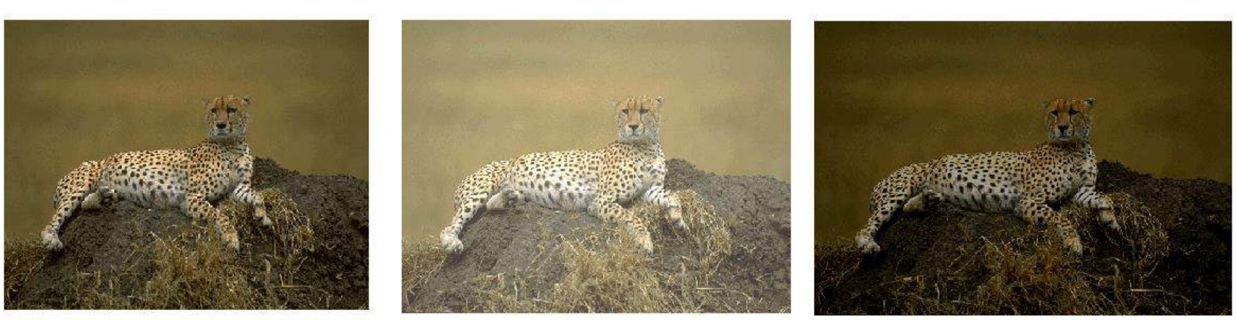


Figure (5) a, b) two frames of video sequence c) their difference

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 (6) a) original image b) image multiplied by 2 c) image divided by 2

3.4 Logical Operation:

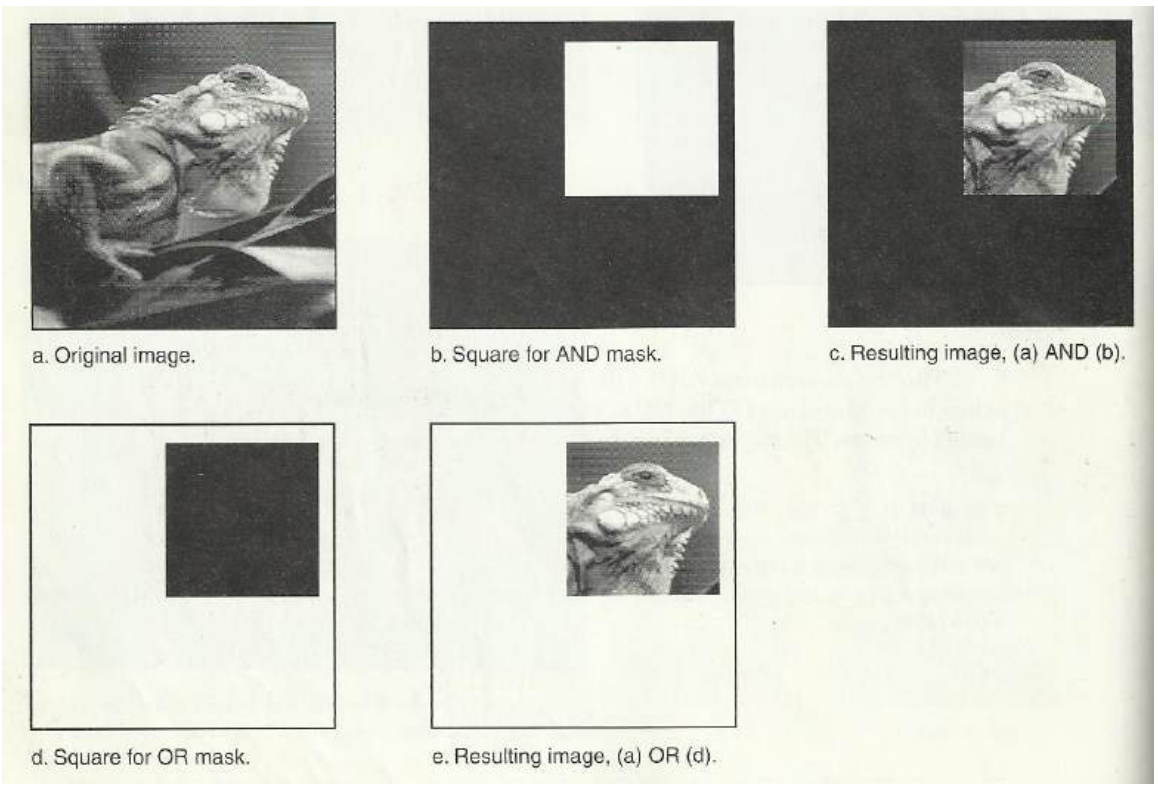
Logical operations apply ***only to binary images***, whereas arithmetic operations apply to multi-valued pixels. Logical operations are basic tools in binary image processing, where they are used for tasks such as ***masking***, ***feature detection***, ***shape analysis*** and **obtain the similarity 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				OR				XOR			
input1	1	1	0	0	1	1	0	0	1	1	0	0
Input2	1	0	1	0	1	0	1	0	1	0	1	0
output	1	0	0	0	1	1	1	0	0	1	1	0

Logical AND & OR operations are useful for the ***masking and compositing*** 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(7) : 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*** white regions of two different images (it required two images).

$$g(x,y) = a(x,y) \wedge b(x,y)$$

Exclusive OR ⊗

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

$$g(x,y) = a(x,y) \otimes b(x,y)$$

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.

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

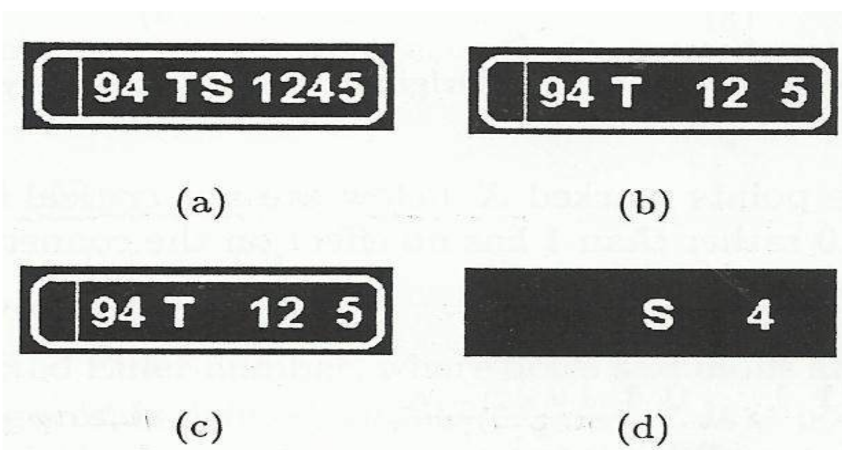


Figure 8) a) input image a(x,y) b) input image b(x,y)

c) a(x,y) ^ b(x,y) d) a(x,y) ⊗(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 though 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 9 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
Endfor
For all pixels coordinates, x and y ,  do
        Increment histogram [ f (x,y) ] by 1
Endfor
```

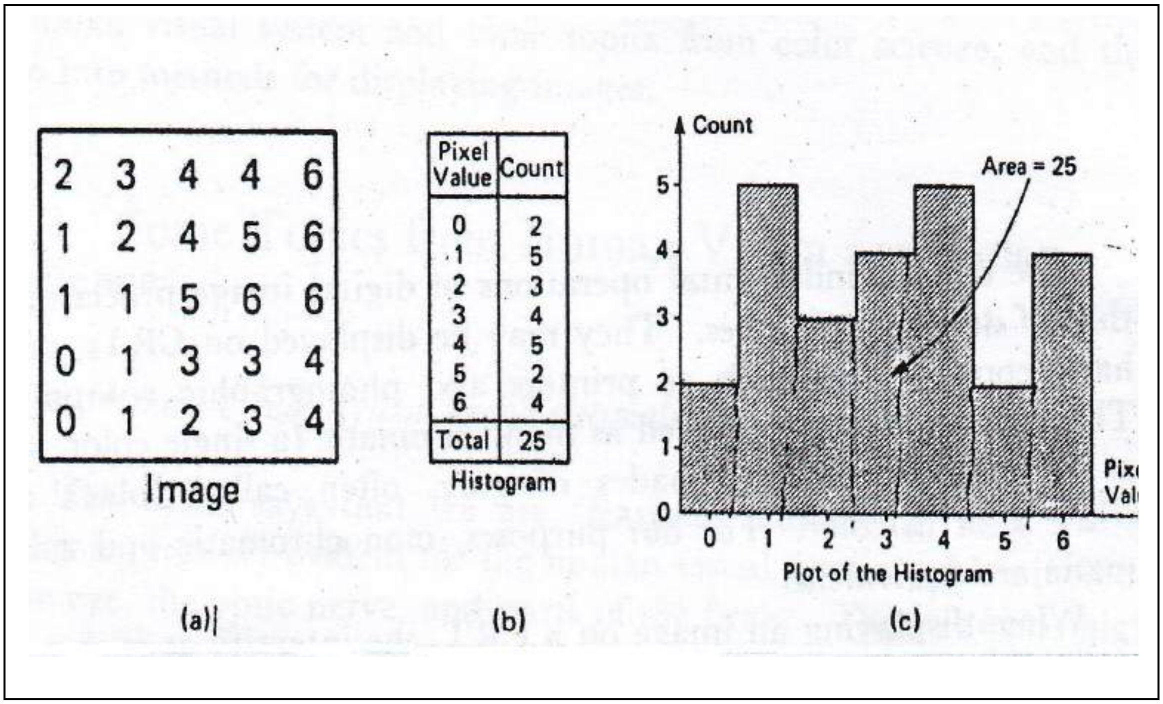


figure 9: sub image and its histogram

The ***shape of the histogram*** provide 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 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.

Probabilistic of 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 . 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 use*** of the histogram is in the selection of ***threshold*** parameter.

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 ***brightnes***s or ***contras***t ***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 **10** 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 a bout 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*** it if a<1 , which can have the effect of merging bins .

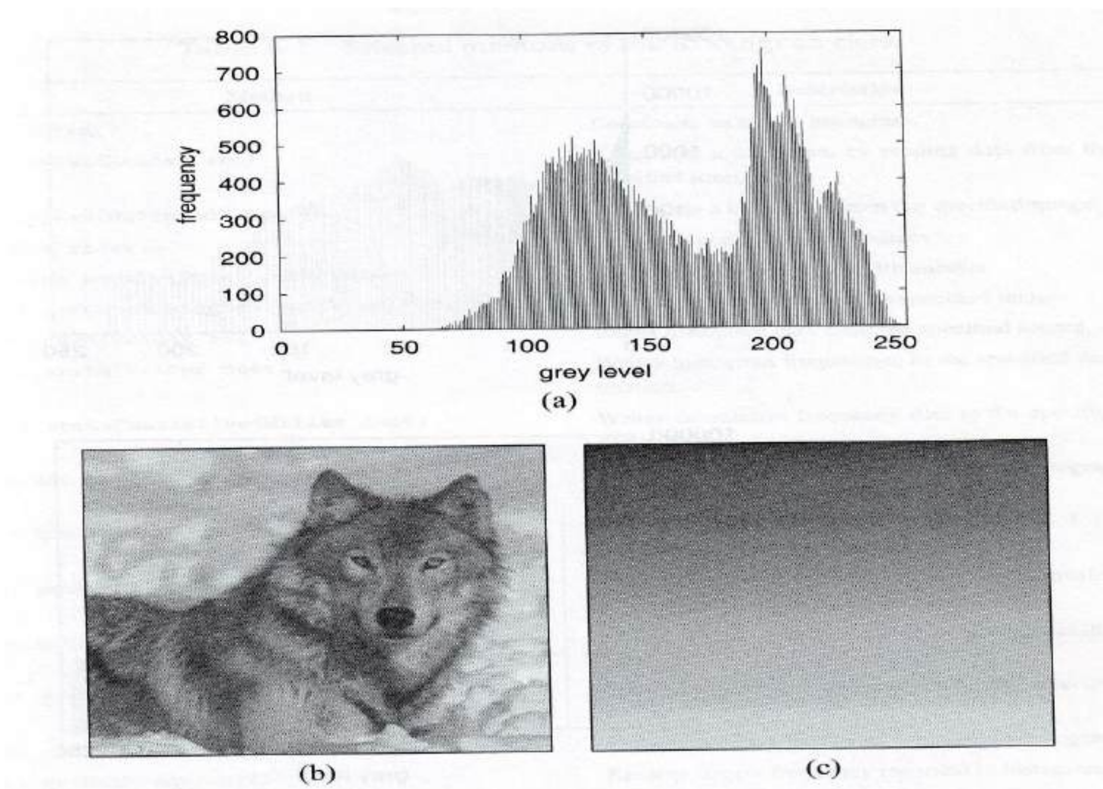


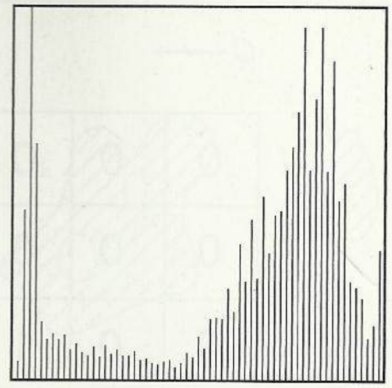
Figure 10: non-uniqueness of a histogram

(a) a histogram (b) an image with (a) as its histogram
(c) a different image with the same histogram

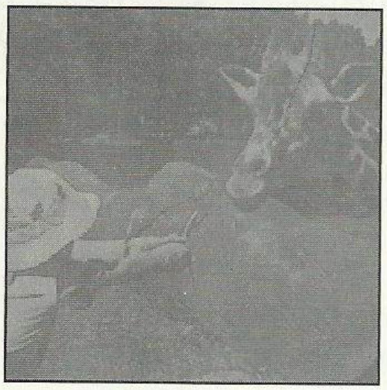
4.2 histogram modification

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 can 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.

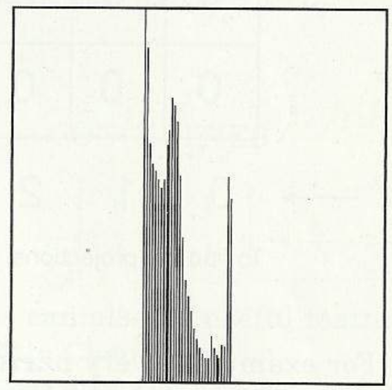
a. Object in contrast with background.



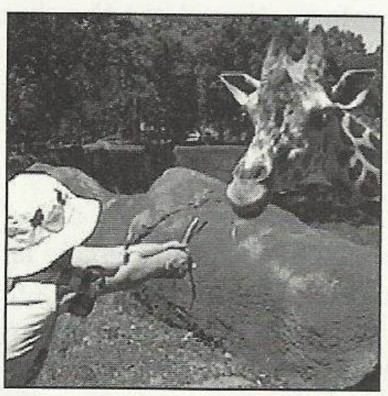
b. Histogram of (a) shows bimodal shape.



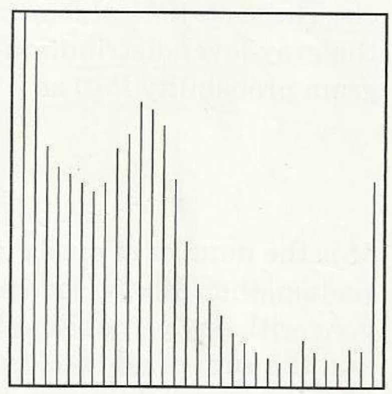
c. Low-contrast image.



d. Histogram of (c) appears clustered.



e. High-contrast image.



f. Histogram of (e) appears spread out

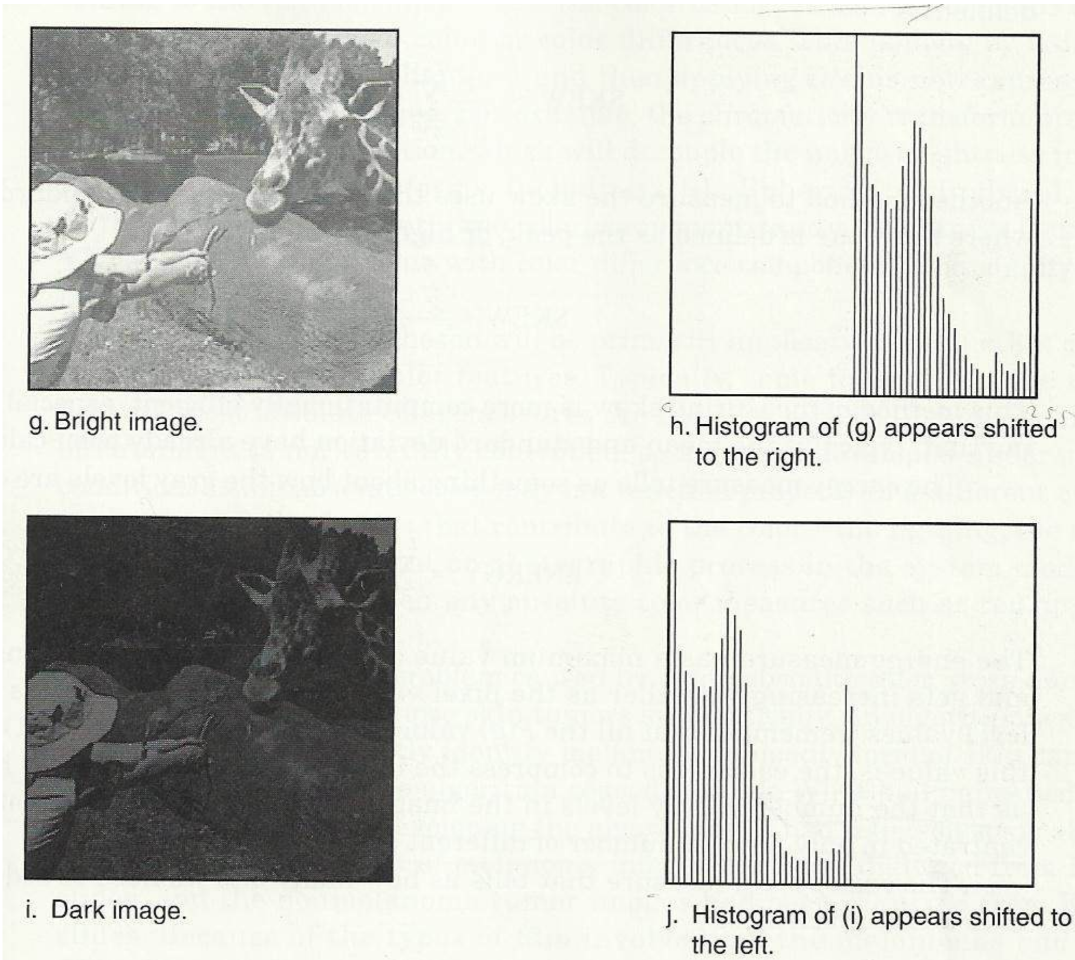


Figure 11 : a variety types of histograms

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 **12** we see a graphical representation of histogram stretch, shrink, and slide.

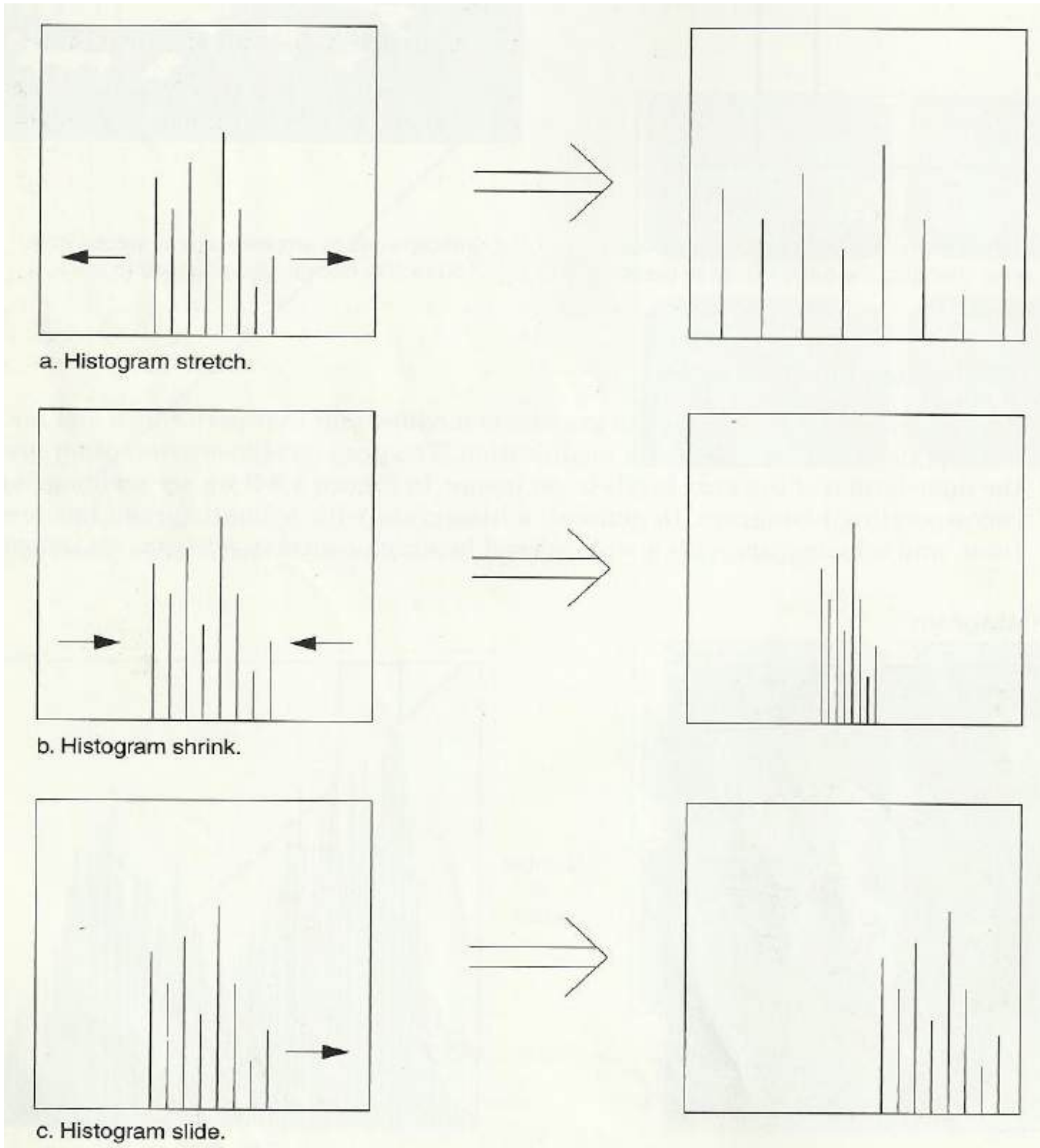


Figure 12 : histogram modification

a. <u>Histogram stretch</u>

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-bitimages 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***.

b. Histogram shrink

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:

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

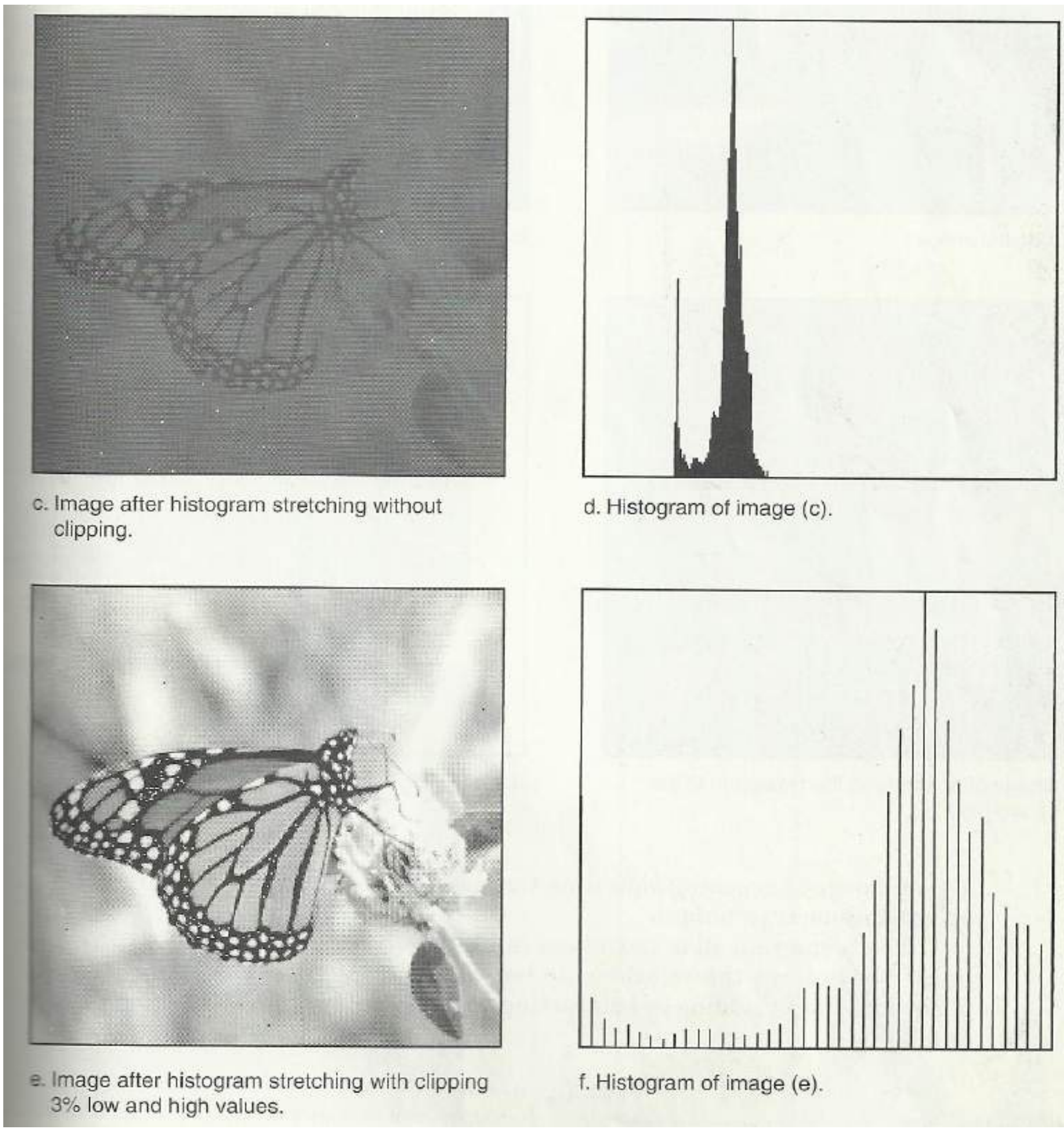
Where :

$I(r,c)_{MAX}$ 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*** an may not seem to be useful as an image enhancement tool.



c. Image after histogram stretching without clipping.

d. Histogram of image (c).

e. Image after histogram stretching with clipping 3% low and high values.

f. Histogram of image (e).

Figure 13: histogram stretch

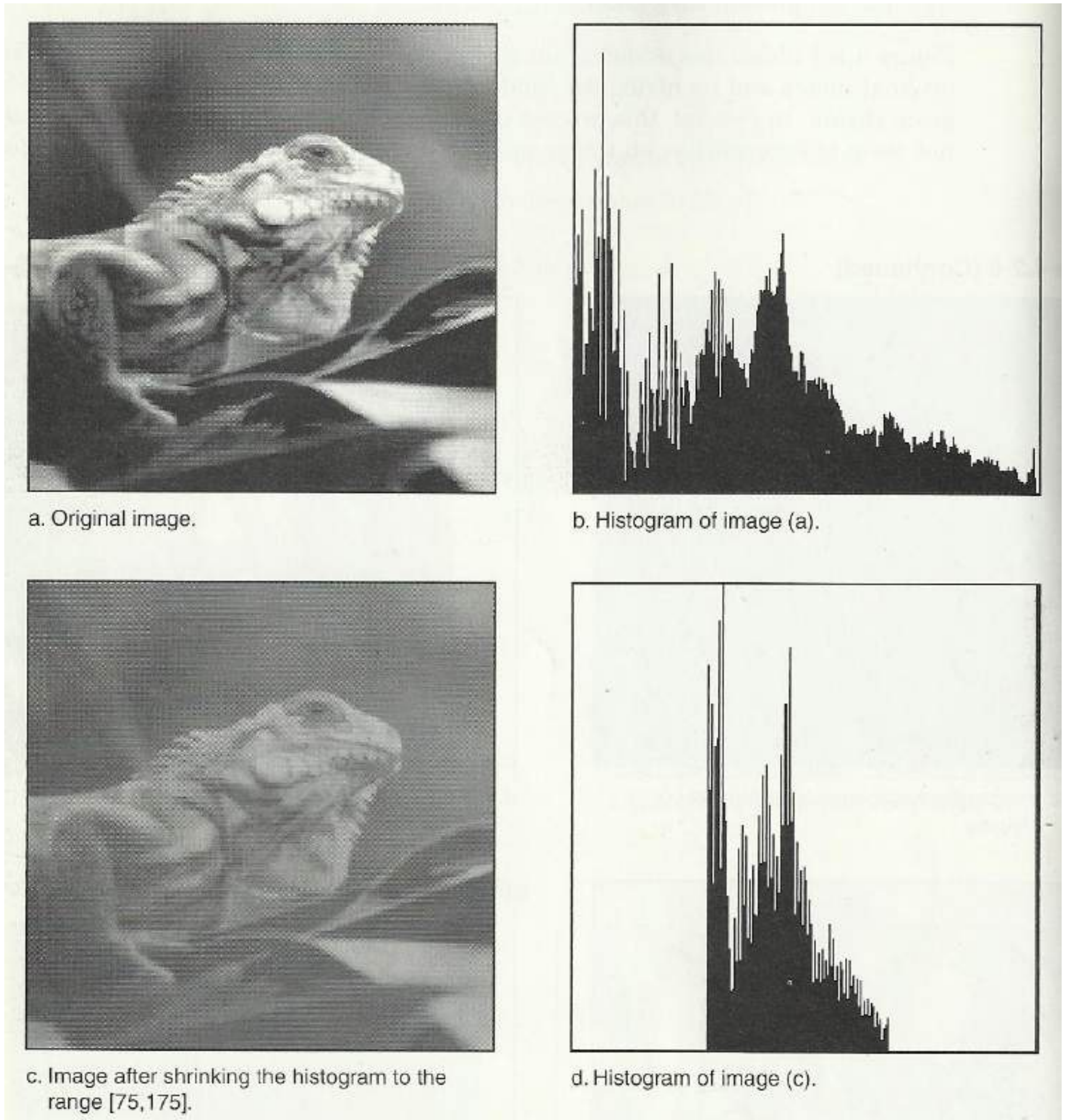


Figure 14: histogram shrink

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.

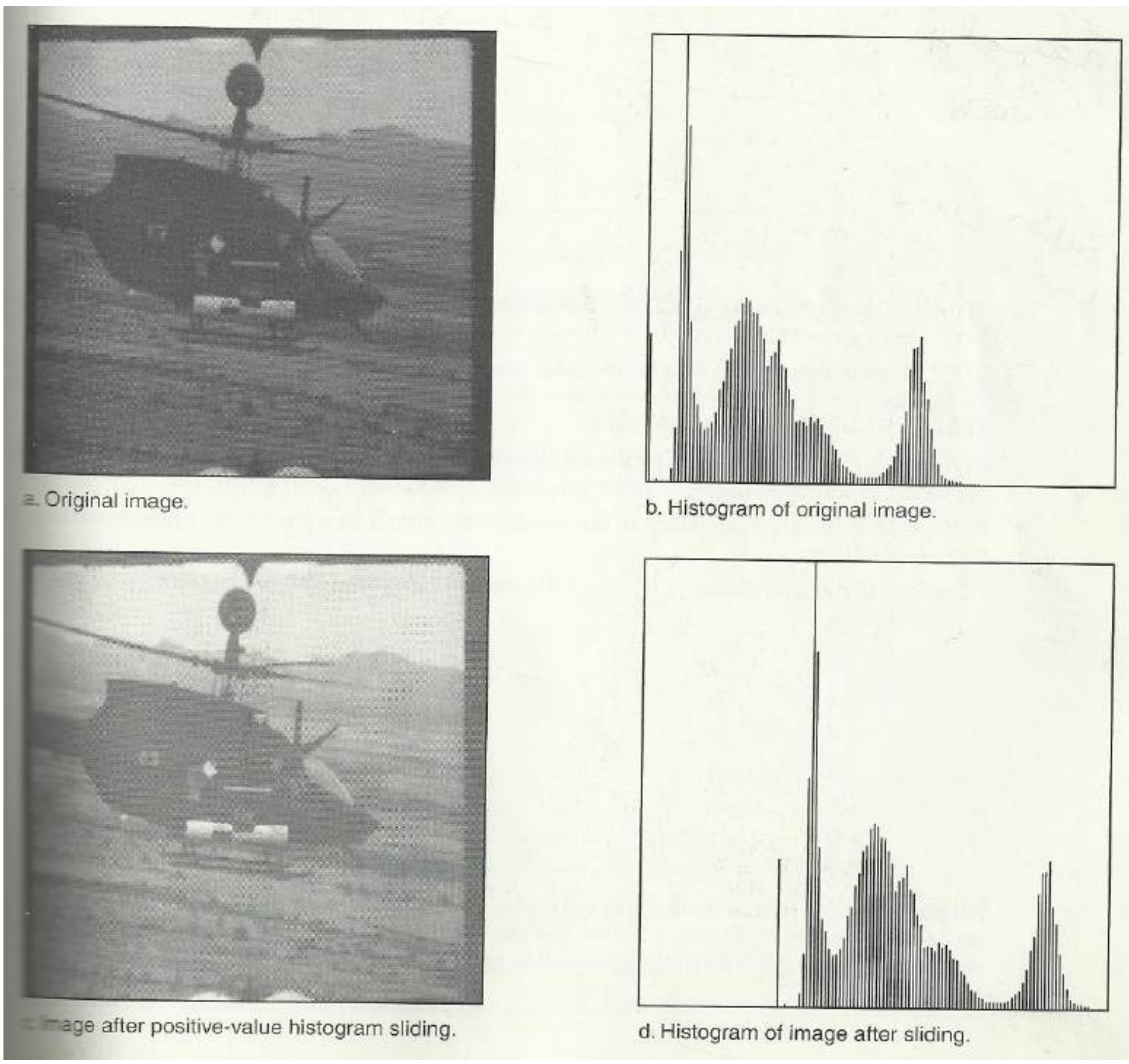


Figure 15: histogram slide

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 are most pixels, fewer gray levels where there are fewer pixels. This tend 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 it's values lie in the range 0-255. The algorithm below shows how this work 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 become 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 it's 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 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

Original Gray level value	Histogram Equalized Gray level values	Number of pixels
0	1	10 (frequency of 0)
1	2	8 (frequency of 1)
2	4	11 (frequency of 2 + 3)
3	4	
4	6	15 (frequency of 4 + 5)
5	6	
6	7	7 (frequency of 6 + 7)
7	7	

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.

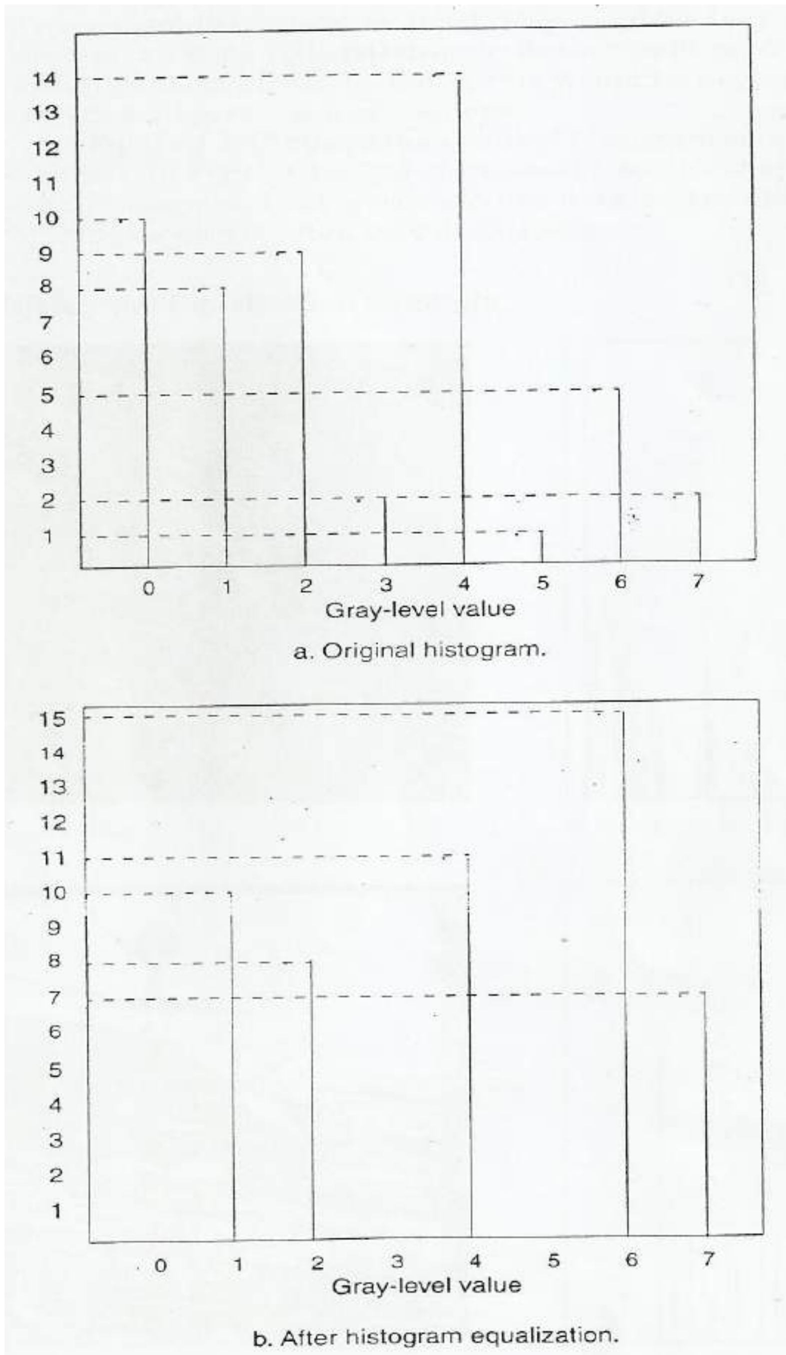


Figure 16 Histogram Equalization

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

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

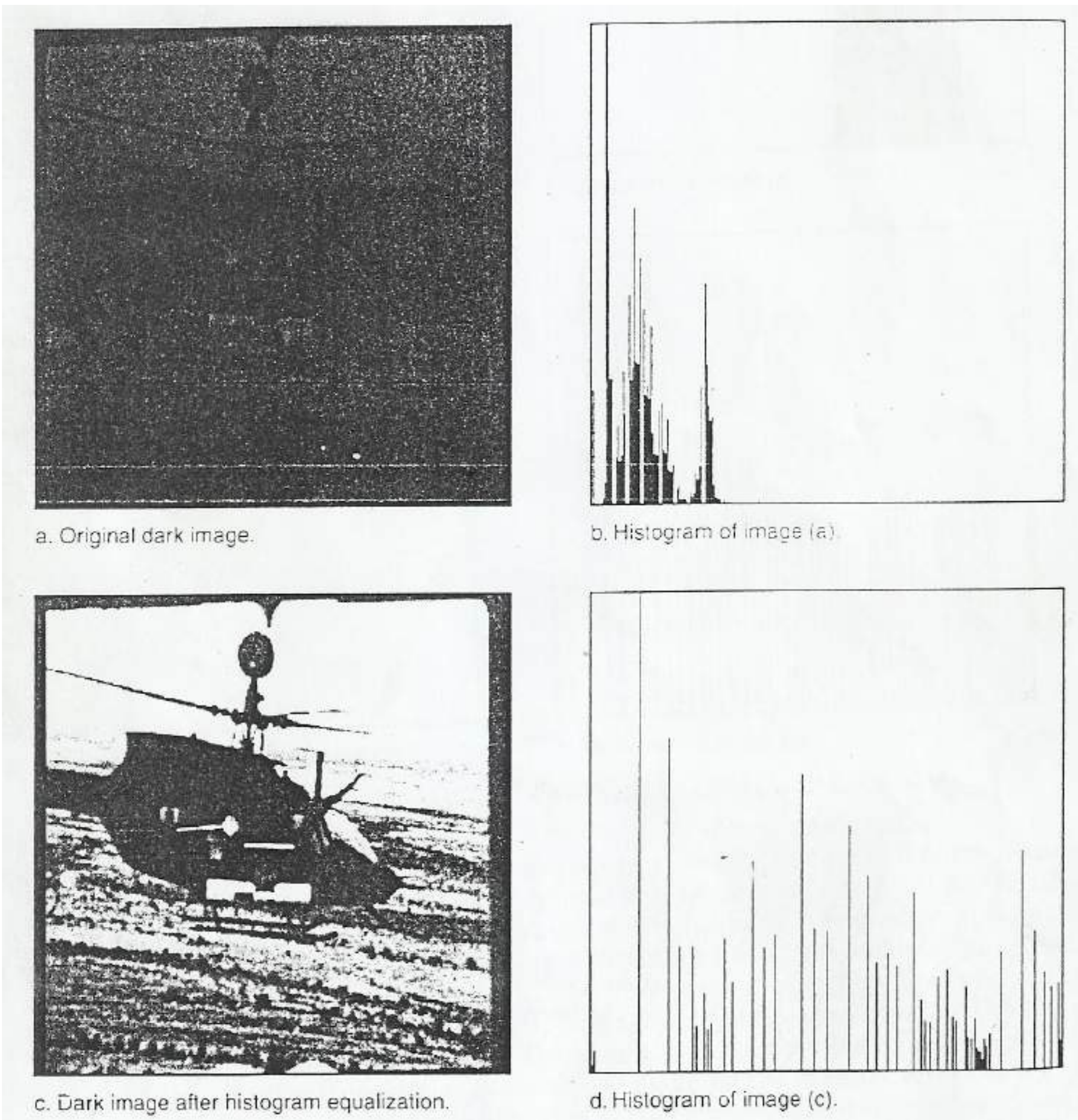


Figure 17 Histogram Equalization example

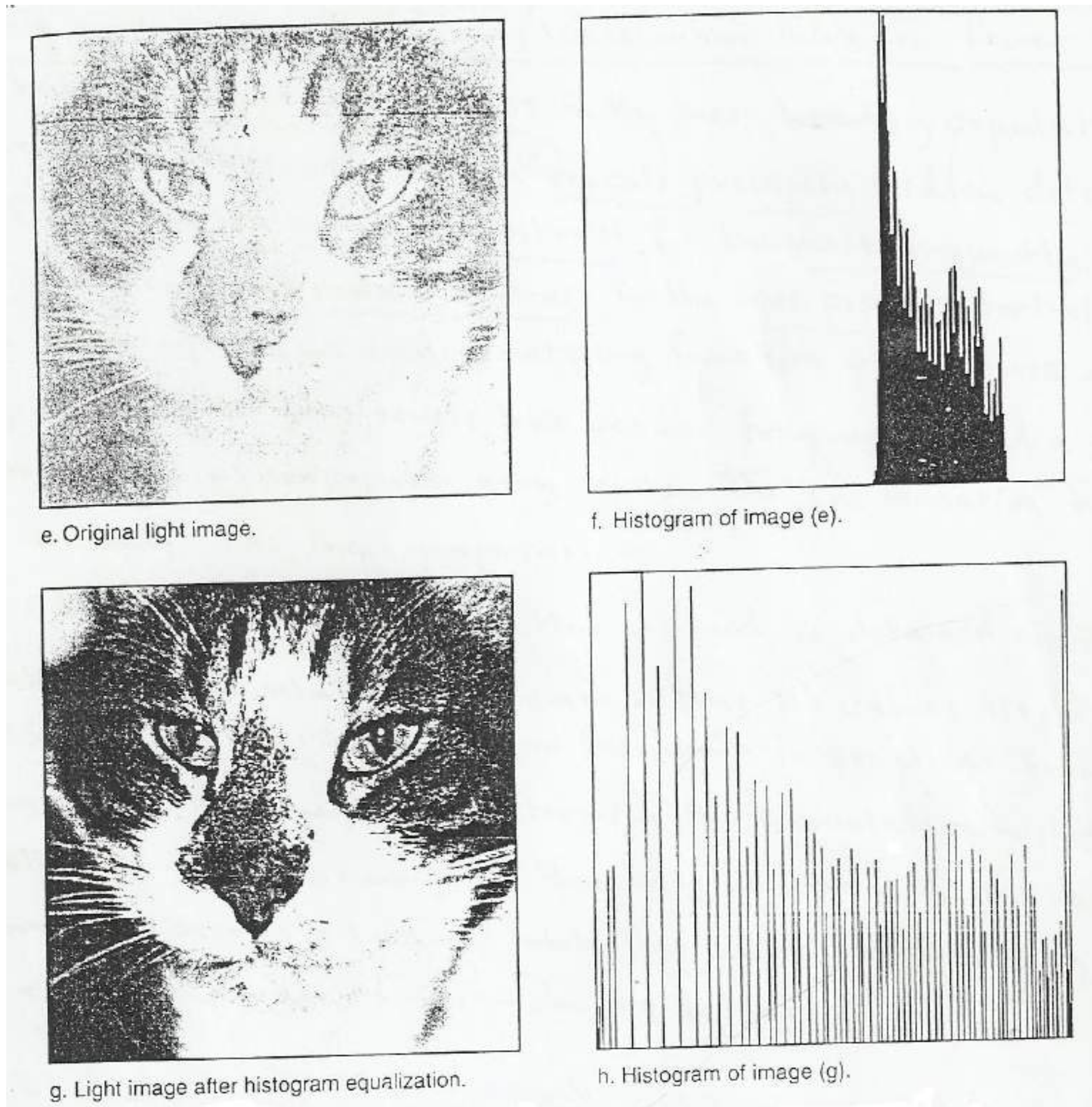


Figure 17 (continue)

5. Image Geometry

Often for image analysis, we want to investigate more closely a specific area within the image, called Region of Interest (ROI). To do this we need operations that ***modify*** the ***spatial coordinates*** of the image, and these are categorized as image geometry operations. The image geometry operations discussed here include ***crop, zoom, enlarge, shrink, translate, and rotate.***

5.1 crop, zoom, enlarge

The image ***crop*** process is the process of selecting a small portion of the image, a sub image, and cutting it away from the rest of the image. After we have cropped a sub image from the original image, we can ***zoom*** in on it by enlarging it. This zoom process can be done in numerous ways, but typically a zero or first order hold is used.

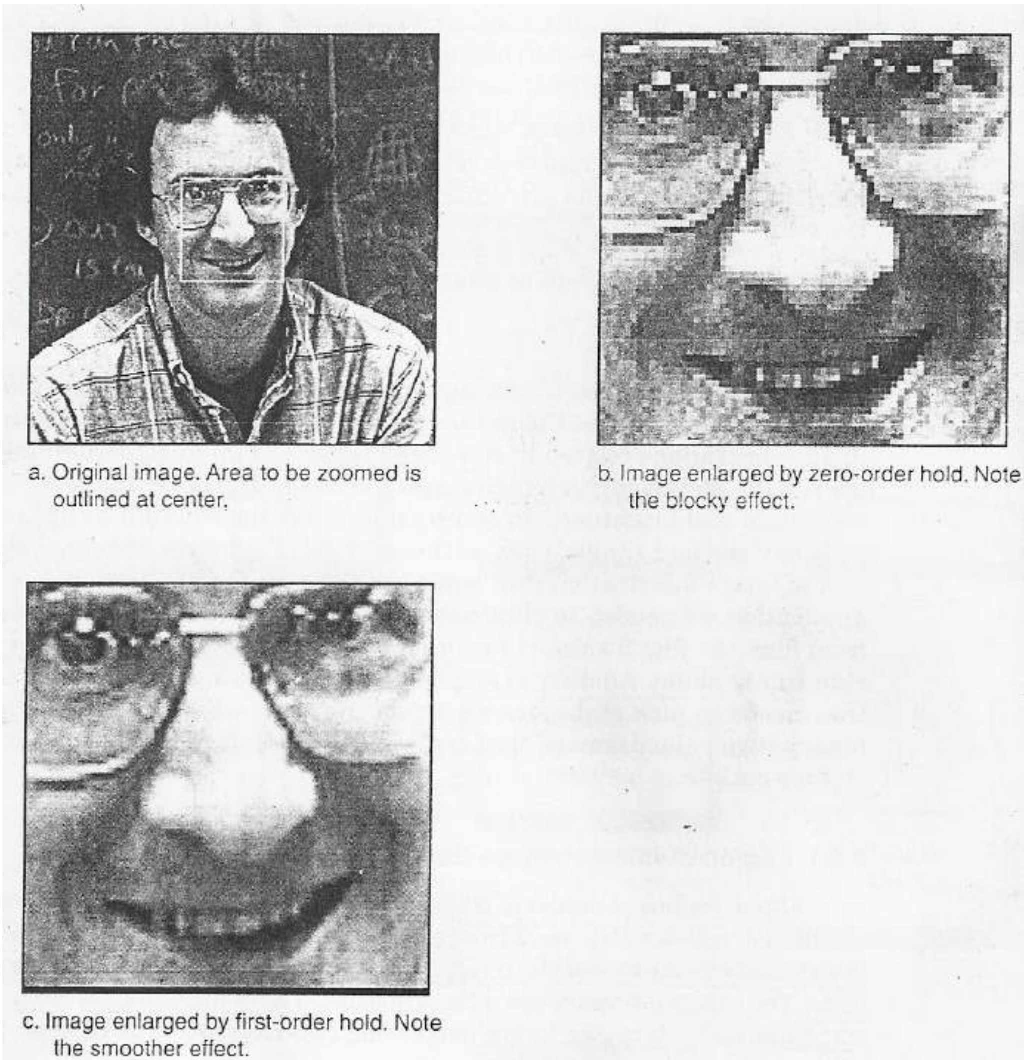
5.1.1 Zero-order hold

A zero-order hold is performed by repeating previous pixel values, thus creating a ***blocky effect***.

5.1.2 First order hold

To extend the image size with a first-order hold , we do linear interpolation between adjacent pixels. A comparison of the images resulting from these two methods is shown in figure 16.

Although the implementation of the zero-order hold is straightforward, the first-order hold is more complicated. It can be done in two different ways



a. Original image. Area to be zoomed is outlined at center.

b. Image enlarged by zero-order hold. Note the blocky effect.

c. Image enlarged by first-order hold. Note the smoother effect.

Figure 18 : zooming methods

a. <u>Averaging method</u>

The easiest way to do this is to find the average value between two pixels and use that as the pixel value between those two pixels; we can do this for the rows first , as follows:

ORIGINAL IMAGE ARRAY

$$\begin{pmatrix} 8 & 4 & 8 \\ 4 & 8 & 4 \\ 8 & 2 & 8 \end{pmatrix}$$

IMAGE WITH ROWS EXPANDED

$$\begin{pmatrix} 8 & 6 & 4 & 6 & 8 \\ 4 & 6 & 8 & 6 & 4 \\ 8 & 5 & 2 & 5 & 8 \end{pmatrix}$$

The first two pixels in the first row are averaged, (8 + 4) / 2 = 6, and this number is inserted between those two pixels. This is done for every pixels pair in each row. Next, take that result and expand the columns in the same way, as follow :

IMAGE WITH ROWS AND COLUMNS EXPANDED

$$\begin{pmatrix} 8 & 6 & 4 & 6 & 8 \\ 6 & 6 & 6 & 6 & 6 \\ 4 & 6 & 8 & 6 & 4 \\ 6 & 5.5 & 5 & 5.5 & 6 \\ 8 & 5 & 2 & 5 & 8 \end{pmatrix}$$

This method allows us to enlarge an $N \times N$ size of $(2N - 1) \times (2N - 1)$ and can be repeated as desired.

b. <u>Convolution method</u>

Another method that can achieves the same result requires a mathematical process called ***convolution***. With this method of image enlargement, a two-step process is required : **1)** extend the image by adding rows and columns of zeros between the existing rows and columns and **2)** perform convolution. The image is extended as follow :

ORIGINAL IMAGE ARRAY IMAGE WITH ROWS EXPANDED

$$\begin{pmatrix} 3 & 5 & 7 \\ 2 & 7 & 6 \\ 3 & 4 & 9 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Next ,we use convolution mask, which is slide across the extended image, and performs a simple arithmetic operation at each pixel location.

CONVOLUTIONMASK FORFIRST ORDER HOLD

$$\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

the convolution process requires us to ***overly*** the mask on the image, ***multiply*** the coincident values , and ***sum*** all these results. This is equivalent to finding the vector inner product of the mask with the underlying sub image. The vector inner product is found by overlying the mask on a sub image, multiplying coincident terms, and summing the resulting products. For example, if we put the mask over the upper-left corner of the image, we obtain (from right to left, and top to bottom) :

$$\frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) + \frac{1}{2}(0) + 1(3) + \frac{1}{2}(0) + \frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) = 3$$

Note that the ***existing image values do not change***. the next step is to slide the mask over by one pixel; and repeat the process, as follows :

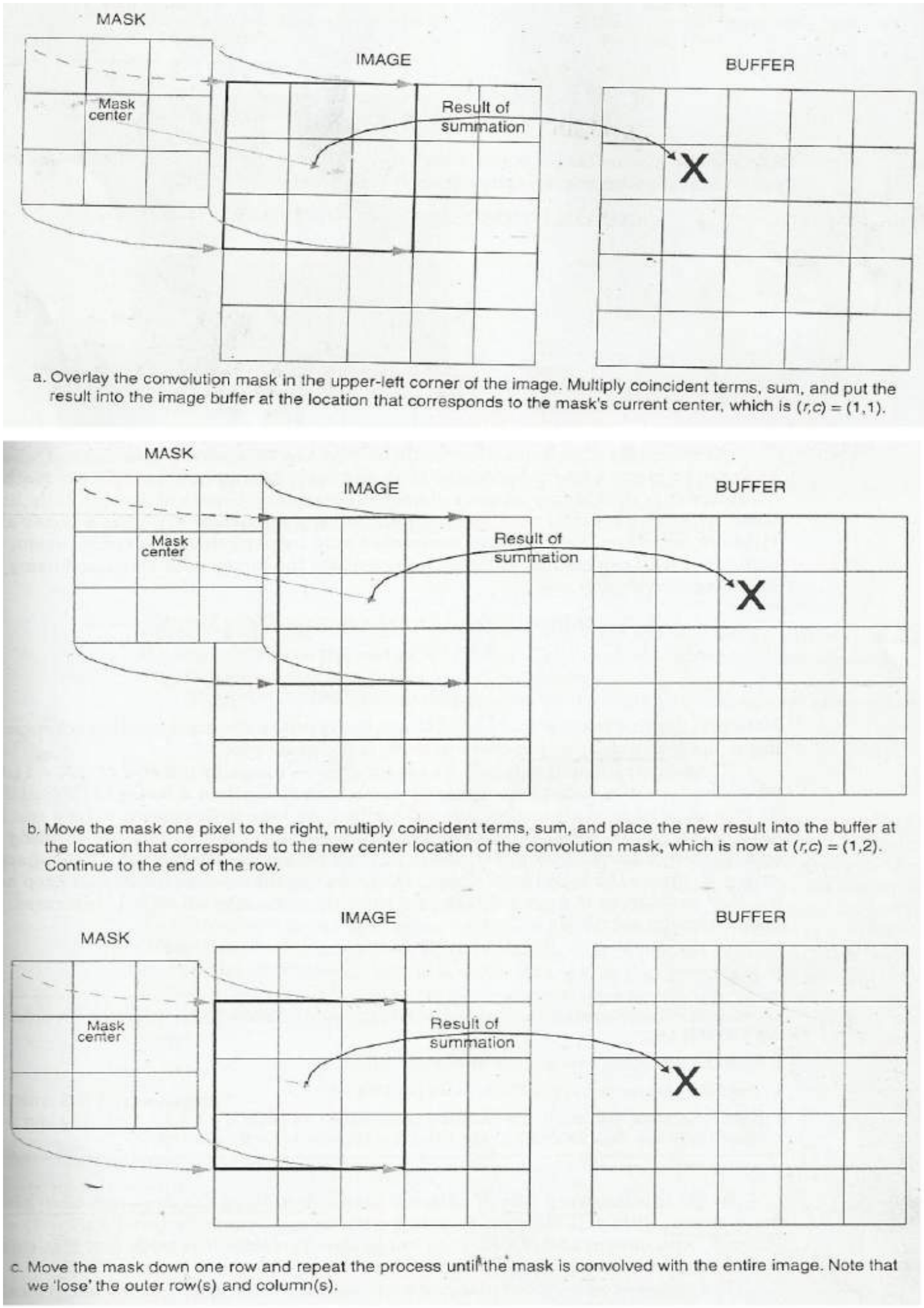
$\frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) + \frac{1}{2}(3) + 1(0) + \frac{1}{2}(5) + \frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) = 4$

Note this is the average of the two existing neighbors. This process continues until we get to the end of the row, each time ***placing*** the ***result*** of the operation in the location corresponding to the ***center of the mask***.

When the end of the row is reached, the mask is moved down one row, and the process is repeated row by row until this procedure has been performed on the entire image; the process of ***sliding, multiplying,*** *and* ***summing*** is called ***convolution*** (see figure 17). Not that the ***output image*** must be put in a separate image array called ***buffe***r, so that the existing values are ***not overwritten*** during the convolution process.

a. <u>Linearly interpolate method</u>

The above two methods will only allow us to enlarge an image by a factor of (2N – 1) × (2 N– 1) , but what if we want to enlarge an image by something ***other than*** a factor of (2N – 1)? To do this we need to apply a more general method; we take two adjacent values and ***linearly interpolate*** more than one value between them. This is done by defining an enlargement number ***K*** and then following this process: **1)** subtract the two adjacent values **2)** divide the result by ***K*** **3)** add that result to the smaller value, and keep adding the result from the second step in a running total until all (***K*** – 1) intermediate pixel locations are filled.



a. Overlay the convolution mask in the upper-left corner of the image. Multiply coincident terms, sum, and put the result into the image buffer at the location that corresponds to the mask's current center, which is $(r,c) = (1,1)$.

b. Move the mask one pixel to the right, multiply coincident terms, sum, and place the new result into the buffer at the location that corresponds to the new center location of the convolution mask, which is now at $(r,c) = (1,2)$. Continue to the end of the row.

c. Move the mask down one row and repeat the process until the mask is convolved with the entire image. Note that we 'lose' the outer row(s) and column(s).

Figure 19 : the convolution process

EXAMPLE

if we want to enlarge an image to ***three times*** its original size, and we have two adjacent pixel values 125 and 140 .

1. Find the difference between the two values 140 – 125 = 15.
2. The desired enlargement factor is ***K*** = 3 , so we get 15 / 3 = 5.
3. Next determine how many intermediate pixel values we need : K – 1 = 3 – 1 = 2. The two pixel values between 125 and 140 are 125 + 5 = 130 and 125 + 2*5 = 135.

We do this for every pair of adjacent pixels, first along the rows and then along the columns. This will allow us to enlarge the image by any factor K (N – 1) + 1, where K is an integer and N × N is the image size. Typically, N is large and K is small, so this is approximately equal to KN.

5.2 Image Shrinking

The process opposite to enlarge an image is shrinking it. This is not typically done to examine a ROI more closely but to reduce the amount of data that needs to be processed.

Image shrinking is accomplished by taking groups of pixels that are spatially adjacent and mapping them to one pixel. This can be done in one of ***three ways:*** 1) averaging 2) median 3) decimation. For averaging method, we take all the pixels in each group (for example 2 × 2 block of pixels) and find the average gray level by summing the values and dividing by the number of pixels in the group. within the second method, median, we sort all the pixels values from lowest to highest and then select the middle value. The third approach, decimation, also known as sub sampling, entails simply eliminating some of the data. For example to reduce the image by a factor of two, we simply take every other row and column and delete them.

EXAMPLE

For the below 2 × 2 block of pixels, shrink this block using the three methods of shrinking.

$$\begin{bmatrix} 6 & 8 \\ 1 & 9 \end{bmatrix}$$

Averaging	**median**	**decimation**
6+8+1+9=24/4=6	1,6,8,9 = (6+8)/2=7	eliminate the first row and the first column and the result is 9

5.3 translation and rotation

Two other operation of interest for the ROI image geometry are translation and rotation. These processes may be performed for many application-specific reasons, for example to align an image with a known template in a pattern matching process or to make certain image details easier to see. The translation process can be done with the following equations :

$$r' = r + r_o$$

$$c' = c + c_o$$

where **r'** and **c'** are the new coordinates, **r** and **c** are the original coordinates, and $\mathbf{r_o}$ and $\mathbf{c_0}$ are the distance to move, or translate, the image.

The rotation process requires the use of these equations :

$$r^{\wedge} = r\ (\cos \Theta) + c\ (\sin \Theta\ )$$

$$c^{\wedge} = -\ r\ (\sin \Theta) + c\ (\cos \Theta\ )$$

where $r^{\wedge}$ and $c^{\wedge}$ are the new coordinate, r and c are the original coordinates, and Θ is the angle to rotate the image. Θ is defined in a ***clockwise direction*** from the horizontal axis at the image origin in the upper-left corner.

The rotation and translation process can be combined into one set of equations :

$$r^{\wedge\prime} = (\ r + r_o)\ (\cos \Theta) + (\ c + c_o)\ (\sin \Theta\ )$$

$$c^{\wedge\prime} = -\ (\ r + r_o)\ (\sin \Theta) + (\ c + c_o)\ (\cos \Theta\ )$$

where $r^{\wedge\prime}$, $c^{\wedge\prime}$ are the new coordinates and r, c, r_o, c_o and Θ are previously defined.

6. Noise

6.1 Preview

Noise is any undesired information that contaminates an image. The ideal situation (***no noise***) never occurs in practice, so there is a little point in ignoring it. Hence, one of the primary concerns of digital image processing is to increase image quality through the moderation of the degradations introduced by the noise which contaminate the image.

Noise cannot be predicted accurately because of its ***random nature***, and is ***characterized only statistically***, it cannot even be measured accurately from noisy image, since the contribution to the gray levels of the ***noise cannot be distinguished from the pixel data.***

All image-acquisition processes are subjected to noise of some type. Noise occurs due to a great ***many factors*** such as light intensity, type of camera and lens, motion, temperature, atmospheric effects, dust, and others. ***It is very unlikely that two pixels that correspond to precisely the same gray level in the scene will have the same level in the image.***

6.2 Type of noise

Noise appears in images from a variety of sources. The ***digital image acquisition*** process which converts an optical images into a continuous electrical signal that is then sampled , is the ***primary*** process ***by which noise appears*** in digital images. At any step in the process there are fluctuations caused by natural phenomena that add a random value to the exact brightness value for a given pixel.

Digital images are corrupted by noise either during image ***acquisition or during image transmission.*** The image acquisition noise is photo electronic noise (for photo electronic sensors) or film-grain (in the case of photography) , in both cases the noise is ***signal dependence***, in which the ***level of the noise value at each point in the image is a function of the gray level value***. Let f(x,y) be the original image that is recorded on a film slide. Let us denote by g(x,y) the image that is observed if the Film is used as a transparency. The observed image is nonlinear transform of the original image, corrupted by multiplicative noise

$$g(x,y) = C\,(f(x,y))^{-\gamma}\; n(x,y) \qquad \ldots\ldots\ldots\ldots\ldots\ (1)$$

the noise process n(x,y) has a log-normal distribution .

In this case (signal dependent noise) it is obvious that the level of the noise value at each point in the image is a function of the grey level there. The grain seen in some photographs is an example of this sort of noise , and it is generally harder to deal with.

Signal-independent noise is a random set of grey levels, statistically independent of the image data; that is, added to the pixels in the image to give the resulting noisy image. This kind of noise occurs when an image is transmitted electronically from one place to another. If A is a perfect image and N is the noise that occurs during transmission , then the final image B is :

$$B = A + N$$

A and N are unrelated to each other .

In typical images the noise can be modeled with either a Gaussian (Normal), uniform, or salt-and-pepper (impulse) distribution. The shape of the distribution of these noise types as a function of grey levels can be modeled as a histogram , as shown below:

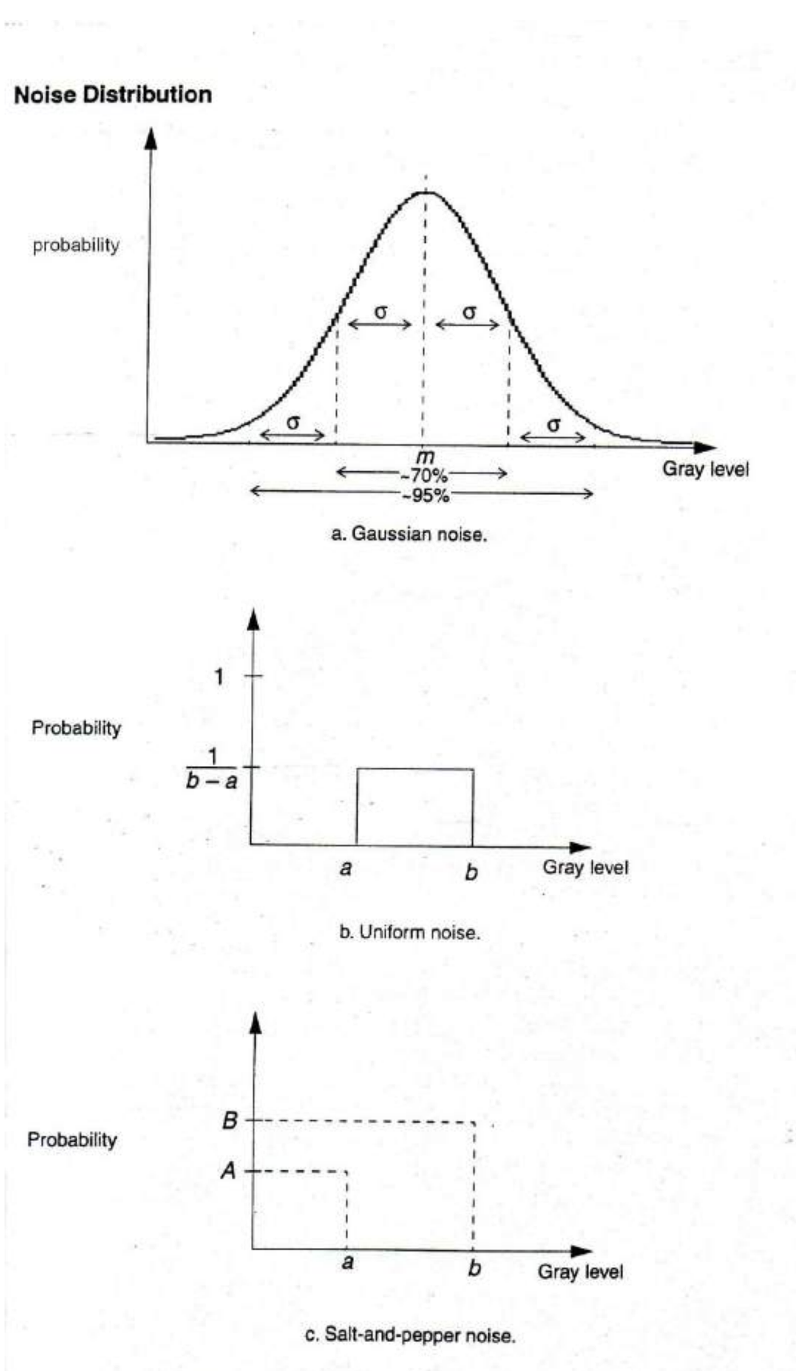


Figure (20) Noise distribution

With the uniform distribution, the grey level values of the noise are ***evenly distributed*** across a specific range, which may be the entire range (0 to 255 for 8-bits), or a smaller portion of the entire range.

In the salt-and-pepper model there are only **two possible values , a and b** , and the probability of each is typically less than 0.1 - with a number greater than this the noise will dominate the image. For a 8-bit image, the typical value for pepper noise is 0 and for salt-noise, 255.

The ***Gaussian*** model is most often used to model natural noise processes, such as those occurring from ***electronic noise*** in the image acquisition system. The ***salt-and-pepper*** type noise is typically caused by ***malfunctioning*** pixel elements in the ***camera sensors, faulty memory locations, or timing errors in the digitization process***. ***Uniform noise*** is ***useful*** because it can be used to ***generate any other type of noise distribution***. Visually , the ***Gaussian and uniform noisy images appear similar, but the image with added salt-and-pepper i***s very distinctive.

6.3 low pass filtering

A low pass filters allows low spatial frequencies to pass unchanged, but suppresses high frequencies.

Low pass filtering creates an image with a smooth appearance because it suppresses any rapidly changing brightness value in the original image . The low pass filters do this by attenuating high spatial frequency information, which corresponds to the rapid change (edge).

Low pass filters (smoothing filters) are designed to reduce the noise, detail, or busyness in an image. Visually they blur the image, although this blur is sometimes considered as enhancement because it impart a softer effect to the image . It is easy to smooth out an image, but the basic problem of smoothing filters is how to do this without blurring out the interesting features .

In general, high frequency information can be removed with a low pass filter, which will have the negative effect blurring the image.

Note that, as the size of the window get bigger, the more information loss occurs, within windows larger than 5×5 the image acquires an artificial "painted" effect, and that's because details (such as lines and corners) smaller than the mask size are eliminated.

Any convolution kernel whose coefficients are all positive will act as a low pass filter . Here are some common spatial convolution masks for low pass filtering.

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

An example of noise reduction by low pass filtering is given by figure (21) . The filtering operation has suppressed, but has not eliminate the noise . It has also blurred the objects of interest, making their edges less well defined .

Note that, the filtering is non-specific, in that it reduce the strength of the high spatial frequency components, irrespective of whether they are due to noise or to meaningful structure in the image.

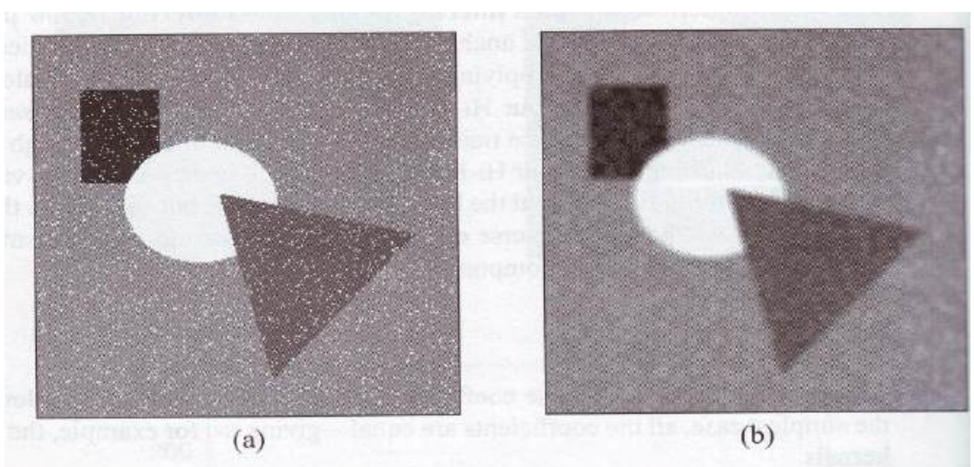


Figure (21) Noise reduction by low pass filtering (a) Synthetic image corrupted by 1% impulse noise (b) Result of applying a 5x5 mean filter

Mean Filter

The mean filter is a low pass linear filter which most commonly used in spatial domain . When all the coefficients of a mask are equal, we can multiply these type of masks by $\frac{1}{N}$ where N is the sum of the mask coefficients, as example, if the first mask multiplied by $\frac{1}{9}$, now, the coefficient sum to one, so convolution with them will not result in an overall brightening of the image . On the other hand, convolution with these kernels is equivalent to computing the mean gray levels over the neighborhood defined by the kernel, for this reason, these kernels are called mean filter . The mean filter work best with Gaussian or uniform noise, this filter blur an image edge or details while mitigating the noise effects .

6.4 Non Linear Filtering - Order Statistical Filters

None linear filter are a large filter class that are based on order statistic, such filters are ***designed to meat various criteria***, for ***example ,edge preservation , adoption to noise statistics or preservation of image details***.

Digital image ***noise usually appears in the high frequencies*** of the image spectrum. Therefore , a low – pass digital filter may be used for noise removal. However, ***linear low-pass filters tend to smear image details (e. g lines, corners),whose power is in the high frequencies as well***. Furthermore , they ***tend to blur the image edge for similar reasons***, thus degrading the digital image quality. ***Nonlinear low-pass filters*** have been proposed in the literature that ***remove noise effectively and preserve image edge and details.***

Convolution is not the only way to of carrying out spatial filtering. None-Linear techniques also exit. A number of these are known collectively as " order statistic filter " or " Rank filters " . typically, these filters operate on small sub images, ***windows,*** and replace the center pixel value (similar to the convolution process). Order statistics is a technique that arranges the neighborhood of a given pixel in sequential order , based on grey level value (from smallest to largest grey- level value) and using this ordering to select a value at particular position, to use it as a new value for the center pixel.

The order statistic filters has ***the advantage of not being kernel-based***. Also, they work very well ***with salt-and-peper*** noise. The order filters are non-linear, so ***their result are some times unpredictable***.

Types of non-linear filtering

(1) Median filter

The most useful of the order filters is the median filters, in which we select the middle pixel value from the ordered set as our output value.

Edge information is very important for human perception. Its preservation and, possibly ,its enhancement is a very important subjective feature of the performance of a digital image filter. Edge, by definition, contain high frequencies. The ***average filter***, which has low-pass characteristics**,** ***smooth*** them and produces images which are unpleasant to the eye.

In contrast, ***the median filter tends to preserve the edge sharpness***. The robustness properties of the median filter make it very suitable for edge filtering.

Median filter is one of the better edge preservation smoothing filters.

The median filter is particularly good at removing certain type of noise, which is the ***impulse noise***. Impulse noise forces a pixels value to one of the extremes of the range- 0 or 255 in the case of 8 bit images. Figure 3-b shows the result of smoothing the noisy image using a 3×3 mean filter . the amplitude of the noise has been reduced, but the image still has a distinctly mottled appearance . furthermore, the main features of interest- the edge- have been blurred. Figure 3-c shows the result of applying 3×3 median filter to the image. The impulse noise has been eliminated completely, and the effect on other feature is minimal .

Note , by applying a median filter, we place the grey levels from the neighborhood in a list, and sort the list into ascending order, that force the noisy values to migrated to the end of the list and therefore do not affect the selection of a new pixel value.

The median is a robust estimator of location, therefore, a signal outlier (e.g. impulse) can have no effect on its performance, even if its magnitude is very large or very small. On the contrary, the average filter is very susceptible to impulses, even a single outlier can destroy its performance. The robustness properties of the median make it very suitable for impulse noise filtering .

Note that, median filtering can eliminate impulse noise only if the noisy pixels occupy less than half the area of the neighborhood .

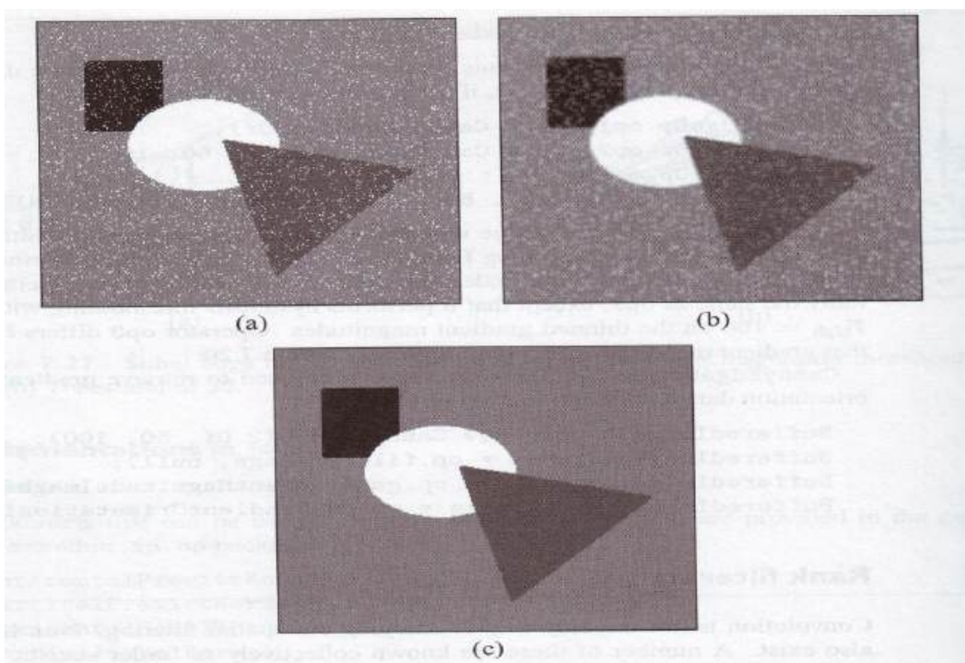


Figure (22) Noise suppression by mean and median filtering (a) Image corrupted by five percent impulse noise (b) Result of 3x3 mean filtering (c) Result of 3x3 median filtering

The ***main disadvantage*** of the median filter, is that it is ***not specific***, any structure that occupies ***less than half of the filters neighborhood*** will tend to be ***eliminated.***

In reality the image have fine details, for example, lines and sharp corners, which are very valuable for human vision. ***These details are usually destroyed by medians having relatively large windows (larger than 5×5).*** It is the ordering process which destroys the structural and spatial neighborhood information.

A median filter can also be used to create a similar smoothing effect, but ***with large mask*** size it creates an almost ***painted*** (and blurred) look.

(2) Minimum and Maximum Filters

the minimum filter is a rank filter in which we select the smallest value within an ordered window of pixel values (i.e the minimum grey level) as the output value. The maximum filter performs similarly , except that we select the largest value within an ordered window of pixel values (i.e. the maximum grey level) as the output value.

Minimum filtering has the effect of ***spreading dark regions*** and contracting bright ones, whereas, maximum filtering has the effect of the ***spreading bright regions*** and contracting dark ones. Both operations have anon linear blurring effecting on the image .

The minimum filter works best when the noise is primarily of the salt-type (high values) , while the maximum filters works best for pepper-type noise (low values) .

On the other hand, ***both fail in the removal of mixed impulse noise***, because minimum and maximum filters tend to enhance the negative and positive spikes respectively. However, cascades of max and min filters can remove mixed impulse noise effectively. However, their performance is generally inferior to that of median filter. Both the maximum and minimum filters have good edge preservation properties. Their ***disadvantage*** that they tend to enhance the bright and dark regions of the image respectively. The main ***advantage*** of maximum and minimum filters, is their computational simplicity and the existence of fast specialized processors for their calculation.

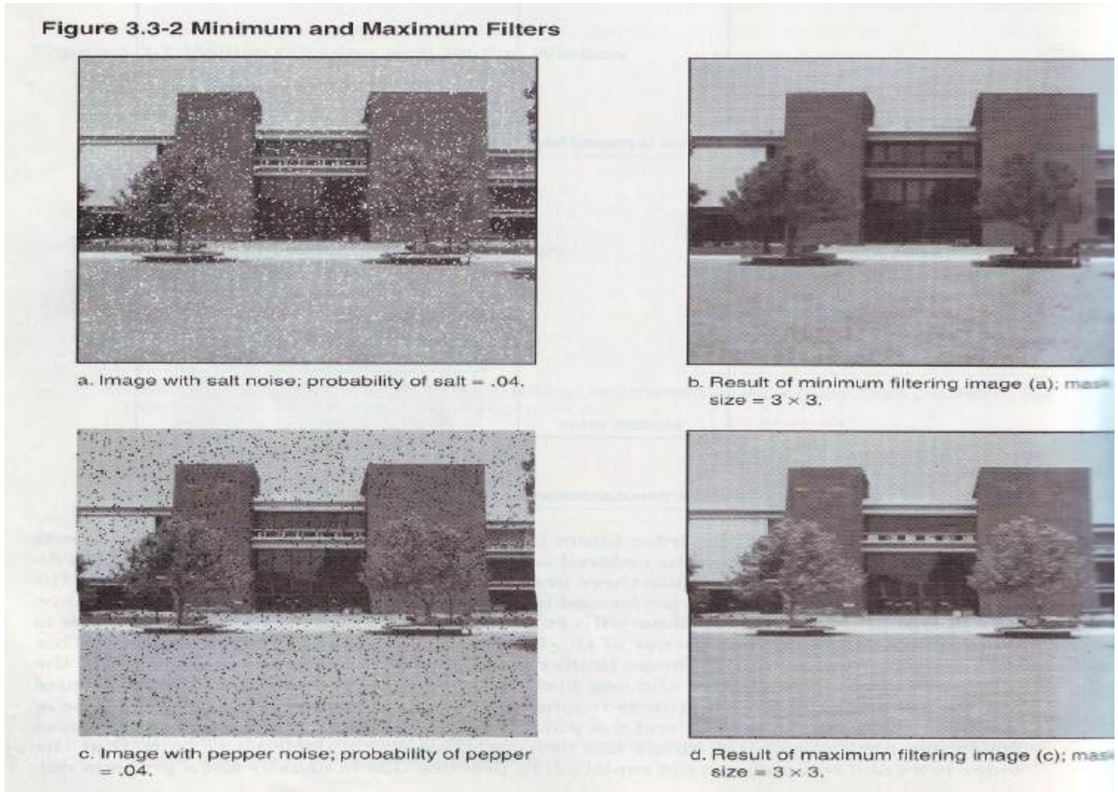


Figure (23) Minimum and Maximum Filters

(3) Mid-point filter

The mid-point filter is actually both order and mean filter because it rely on ordering the pixel values, but it then calculated by an averaging process. The mid-point filter is the average of the maximum and minimum within the window , as follows :

Order set : $I_1 \leq I_2 \leq \ldots.. \leq I_N$

$$Mid - point = \frac{I_1 + I_N}{2}$$

The mid-point is the most useful for Gaussian and uniform noise.

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.

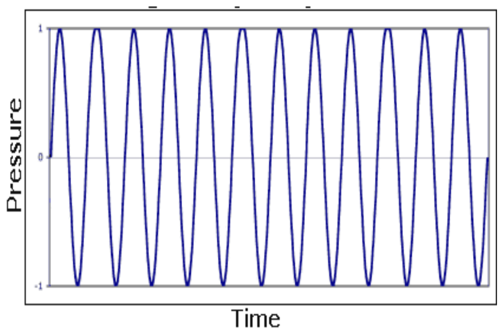
Digital signals : discrete pulses, either on or off, that convey information in a binary form that can be clearly interpreted by computer.

Analog Signals

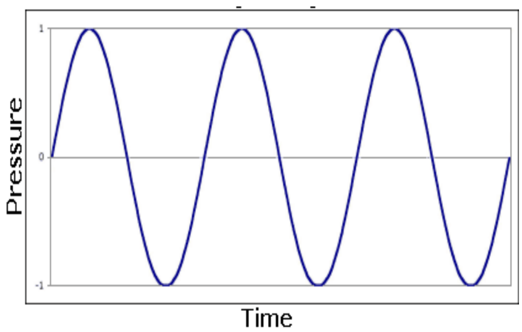
Frequency : the number of cycles per second.

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

Units : Hertz (Hz).



High frequency



Low frequency

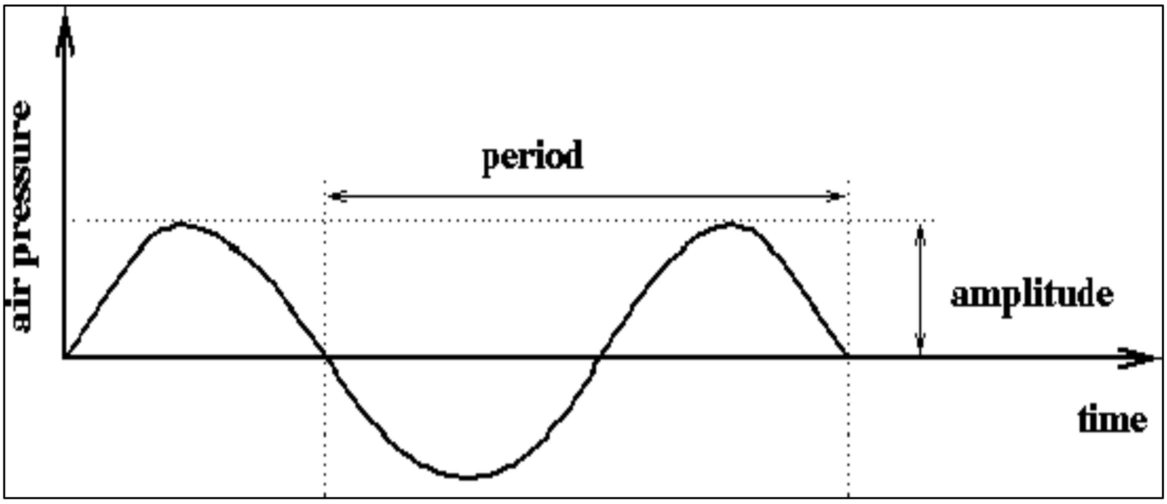
Period : refers to the amount of time, in second, a signal needs to complete one cycle.

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

Amplitude : The amplitude of wave is represent 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.

Units : 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*** subtract the lowest frequency from the highest frequency of the range.

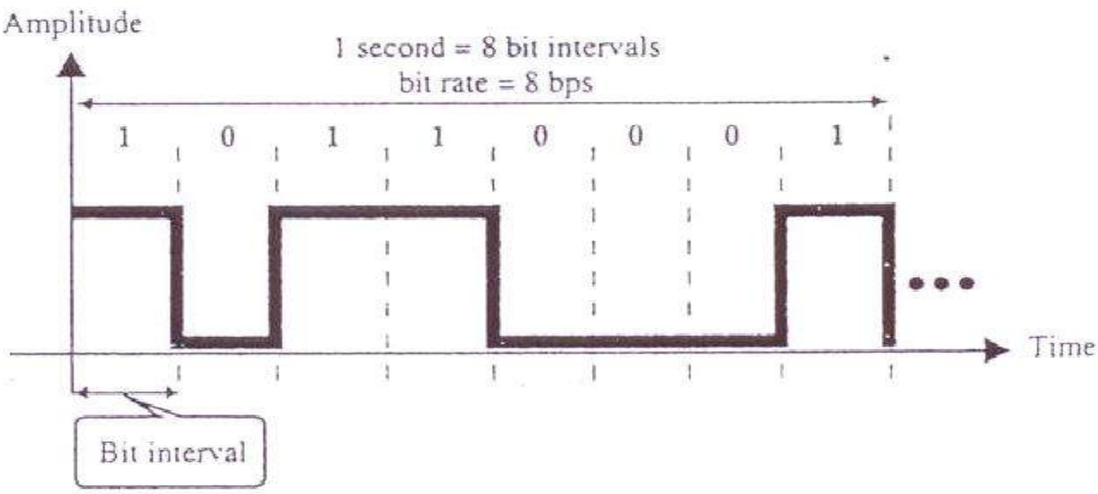
Digital Signals

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.

Basic of Digital Audio

Audio signals is 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.***

1.2 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.

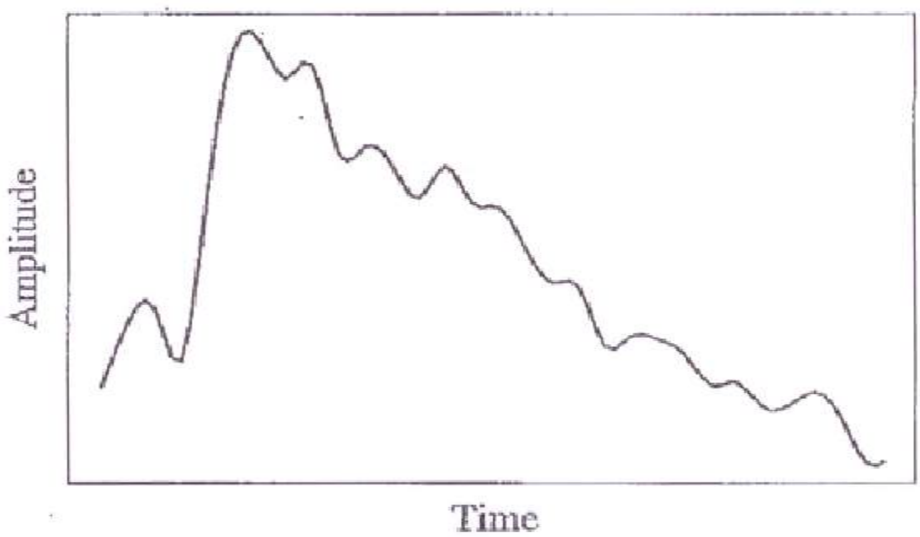


Figure (1) an analog signal : continuous measurement of pressure wave.

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.

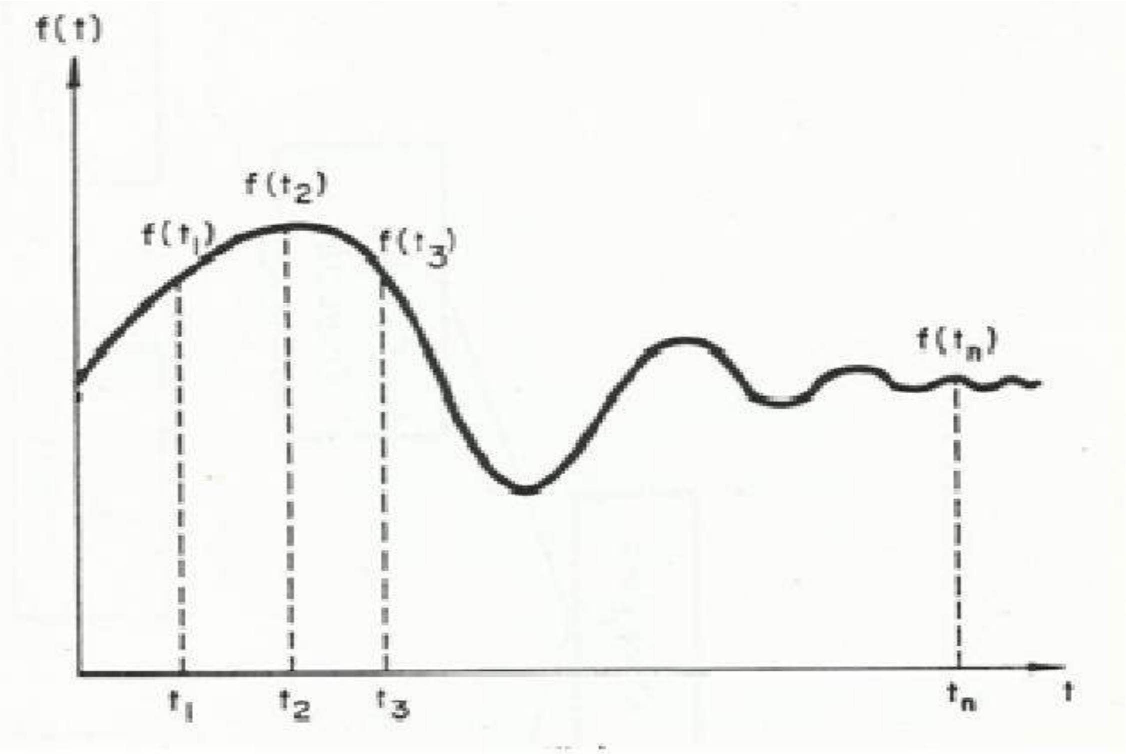


Figure (2) sampling of analog signal

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)?

1.3 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 = $2 \times (600) = 1200$ samples/Second

1.4 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

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

Sol

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

1.5 Audio Filtering

Prior to sampling and AD (analog-to-digital) conversion, the audio signal is also usually filtered to remove unwanted frequencies. the frequencies kept depend on the application. For ***speech***, typically from **50 Hz to 10 KHz** is retained. Other frequencies outside this range are ***blocked by band-pass filter***, which screens out lower and higher frequencies.

An ***audio music*** signals will typically contain from **20 Hz to 20 KHz** (twenty KHz is about the highest squeak we can hear), so the ***band-pass*** filter for music will ***screen out frequencies outside this range***.

At the DA (digital-to-analog) converter end, even though we have removed high frequencies that are likely just noise in any event, they reappear in the output.

The ***reason*** is that because of sampling and then quantization, we have effectively replaced a perhaps ***smooth input signal*** by a ***series of step functions.*** In theory, such a discontinuous signal contains all possible frequencies. Therefore, at the decoder side, a ***low-pass*** filter is used after the DA circuit, making use of the same ***cutoff*** as at the ***high-frequency*** end of the coder's ***band-pass*** filter.

1.6 Audio Quality versus Data Rate

The uncompressed data rate increases as more bits are used for quantization. Stereo information, as opposed to mono, ***doubles the amount of bandwidth*** (in bit per second) needed to transmit a digital audio signal. Table (1) shows how audio quality is related to data rate and bandwidth.

TABLE 1 : Data rate and bandwidth in sample audio applications

Quality	Sample rate (kHz)	Bits per sample	Mono/ stereo	Data rate (if uncompressed) (kB/sec)	Frequency band (Hz)
Telephone	8	8	Mono	8	200–3,400
AM radio	11.025	8	Mono	11.0	100–5,500
FM radio	22.05	16	Stereo	88.2	20–11,000
CD	44.1	16	Stereo	176.4	5–20,000
DAT	48	16	Stereo	192.0	5–20,000
DVD audio	192 (max)	24 (max)	Up to 6 channels	1,200.0 (max)	0–96,000 (max)

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 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.

1.7 Quantization and Transmission of Audio

Audio is analog – the waves we hear travel through the air to reach our eardrums. We know that the basic techniques for **creating digital signal from analog ones consist of sampling and quantization**. ***Sampling*** is invariably done ***uniformly*** – we select a sampling rate and produce one value for each sampling time.

In the magnitude (amplitude) direction, we ***digitize by quantization***, selecting breakpoints in magnitude and remapping any value within an interval to one representative output level.

To be transmitted, sampled audio information must be digitized, and here we look at some of the details of this process. Once the information has been quantized, it can then be ***transmitted or stored.***

1.7.1 Analog-to-Digital Converter (ADC)

ADC : is an electronic integrated circuit which transforms a signal from analog (continuous) to digital (discrete) form. Digital signals only have two states. For digital computer, we refer to binary states, 0 and 1.

1.7.2 Encoding of Audio

Analog-to-digital encoding is the representation of analog information by a digital signal. To record a singer's voice onto a compact disk, for example, you use digital means to replicate analog information. To do so you need to reduce the potentially infinite number of values in an analog message so that they can be represented as a digital stream with a minimum loss of information. Figure (3) shows the analog-to-digital encode.

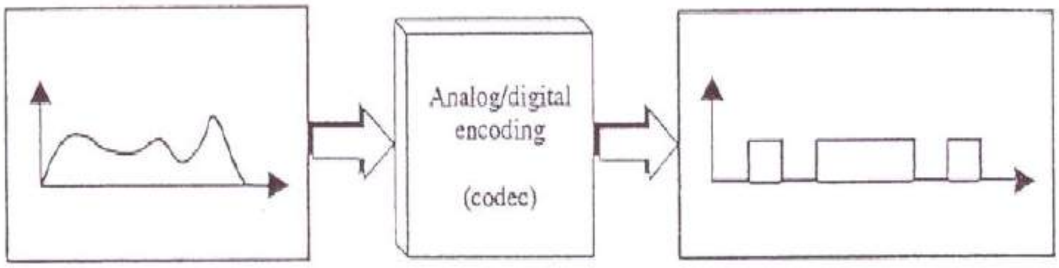


Figure (3) analog-to-digital encoding

In an analog to digital encoding, we are representing the information contained in a continuous wave form as a series of digital pulses (1s or 0s), the problem is how to translate information from an infinite number of values to a discrete numbers without sacrificing sense of quality.

1.7.3 Pulse Amplitude Modulation (PAM)

The ***first step in analog-to-digital encoding*** is called pulse amplitude modulation (PAM). This technique takes analog information, samples it, and generates a series of pulses based on the results of sampling. The term ***sampling*** means measuring the amplitude of the signal at equal intervals (at evenly spaced interval). PAM is the foundation of an important analog-to-digital encoding method called pulse code modulation (PCM). In PAM, the original signal is sampled at equal intervals as shown in figure (4).

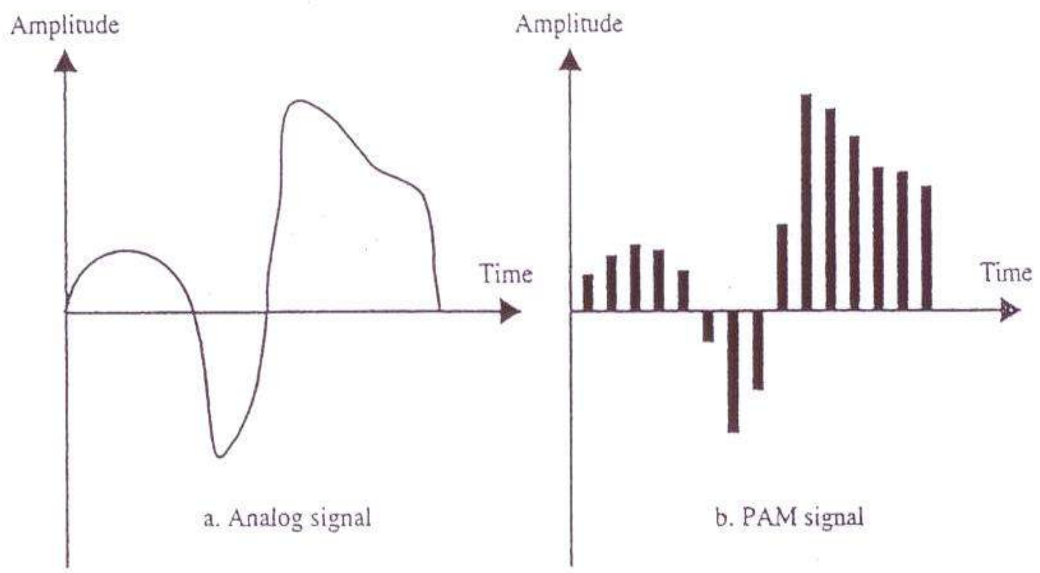


Figure (4) PAM

The method of sampling suing PAM is more useful to other areas of electrical engineering than it is to data communication. The reason PAM is not useful to data communications is that , although it translates the original wave form to a series of pulses, these pulses are still of any amplitude (still an analog signals, not digital).to make them digital, we must modify them by using pulse code modulation(PCM).

1.7.4 Pulse Code Modulation (PCM)

PCM modifies the pulses created by PAM to create a completely digital signals. To do so, PCM first quantizes the PAM pulses. Quantization is a method of assigning integral values in a specific range to sampled instances. The result of quantization is presented in figure (5).

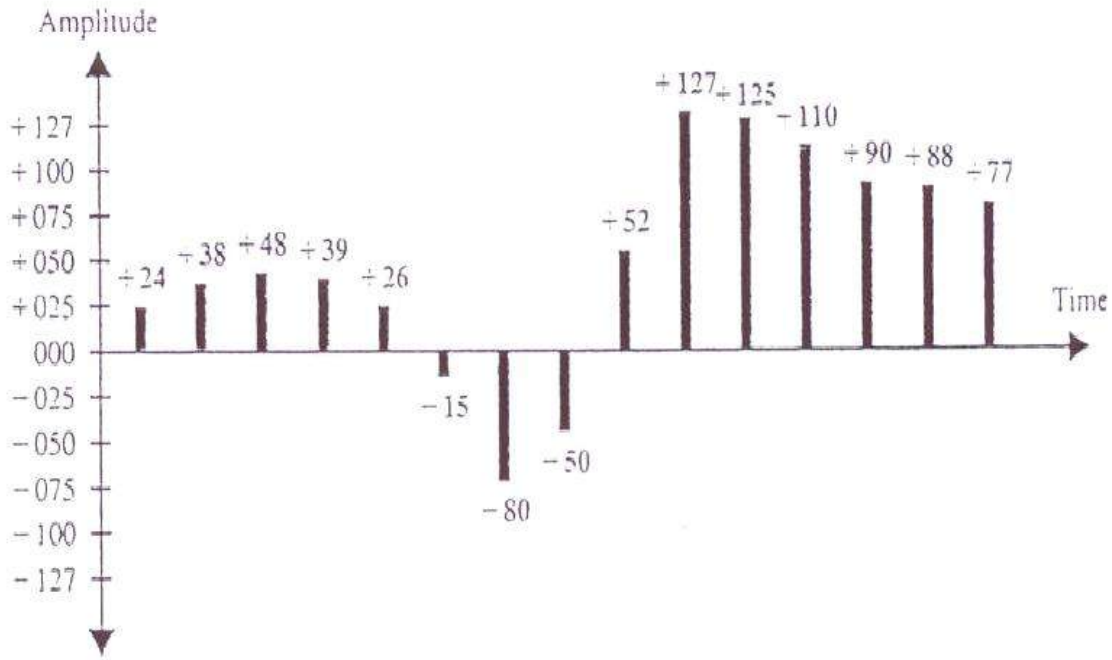


Figure (5) Quantized PAM signal

Figure (6) shows a simple method of assigning sign and magnitude values to quantizated samples. Each value is translated into its ***seven-bit*** binary equivalent. The ***eighth bit*** indicates the sign.

The binary digits are then transformed into a digital signal using one of the digital-to-digital encoding techniques. Figure (7) shows the result of the pulse code modulation of the original signal encoded finally into unipolar signal. Only the first three sampled values are shown.

PCM is actually made of ***four separate processes*** : PAM, quantization, binary encoding, and digital-to-digital encoding. figure (8) shows the entire process in graphic form. PCM is the sampling method used to digitize voice in T-line transmission in the North American telecommunication system.

Figure **6** *Quantizing using sign and magnitude*

+024	00011000	−015	10001111	+125	01111101
+038	00100110	−080	11010000	+110	01101110
+048	00110000	−050	10110010	+090	01011010
+039	00100111	+052	00110110	+088	01011000
+026	00011010	+127	01111111	+077	01001101

Sign bit
+ is 0 − is 1

Figure **7** *PCM*

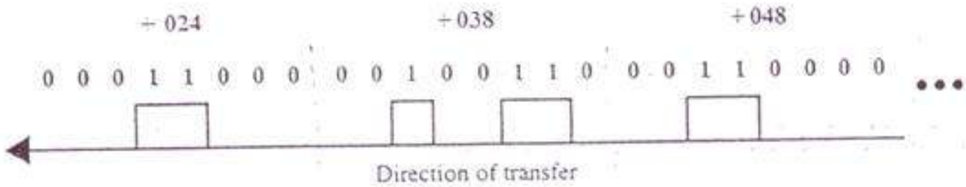
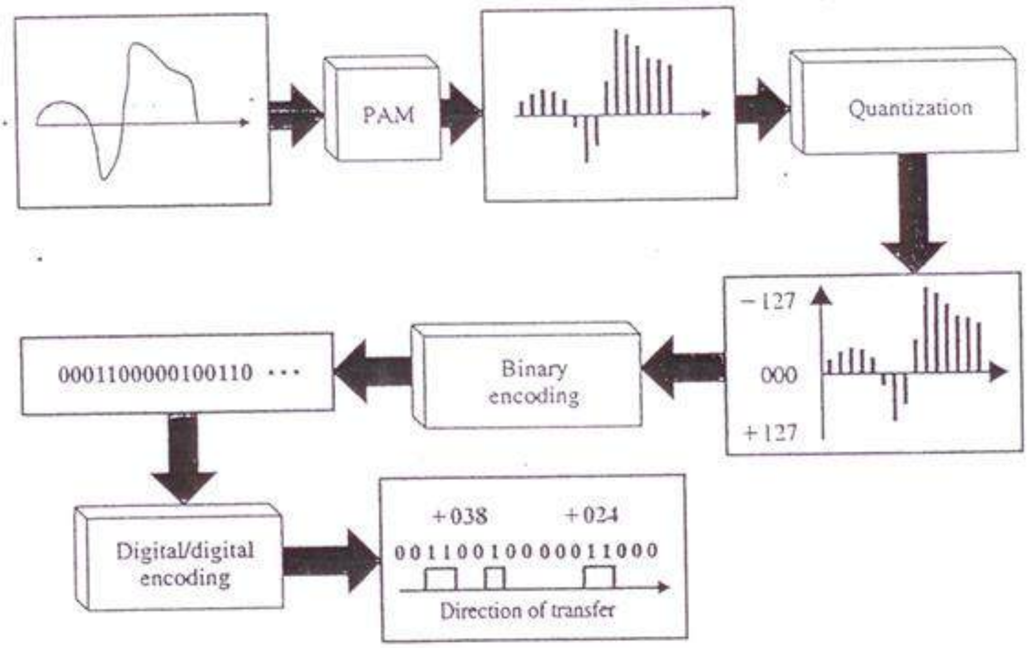


Figure **8** *From analog signal to PCM digital code*



Ex

If the maximum voltage value of PCM signal is + 31 and the minimum value is – 31 , how many bits were used for coding (bit depth).

Sol

5 bits is required for representing the signal value and 1 bit to represent the sing.

The total number of bits required is 6 bits.

1.8 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***.

1.9 Differential Coding of Audio

Audio is often stored not in simple PCM but in a form that exploits differences. For a start, differences will generally be smaller numbers and hence offer the possibility of using fewer bits to store.

An advantage of forming differences is that the histogram of difference signal is usually considerably more peaked than the histogram for the original signal.

Generally, if a time-dependent signal has some consistency over time (temporal redundancy), the difference signal – ***subtracting the current sample from the previous one*** – will have a more peaked histogram, with a maximum around zero. Consequently , if we then go on to assign bit string code words to differences, we can assign short codes to prevalent values and ***long code words*** to rarely occurring ones.

To begin with, consider a lossless version of this scheme. Loss arises when we quantize. If we apply no quantization, we can still have compression – via the decrease in the variance of values that occurs in differences, compared to original signals.

Introduction to Digital Video

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 .

1. 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 the Adobe Flash Player.
Windows Media file	.asf	**Advanced Streaming Format** This file format stores synchronized multimedia data and can be used 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 pictures in Microsoft Resource Interchange File Format (RIFF) format. It is one of the most common formats because audio or video content that is compressed with a wide variety of **codecs** can be stored in an .avi file.
Movie file	.mpg or .mpeg	**Moving Picture Experts Group** This is an evolving set of standards for video and audio compression developed by the Moving Picture Experts Group. This file format was designed specifically for use with Video-CD and CD-i media.
Windows Media Video file	.wmv	**Windows Media Video** This file format compresses audio and video by using the Windows Media Video **codec**, a tightly compressed format that requires a minimal amount of storage space on your computer's hard disk.

2. Color Models in Video

methods of dealing with color in digital video derive largely from older analogue methods of coding color for TV. Typically , some version of Luminance Is combined with color information in a single signal. For example, a matrix transform method called YIQ is used to transmit TV signals in North America and Japan. This coding also makes it way into VHS video Tape coding in these countries, since video tape technologies also use YIQ.

In Europe, the video tape uses PAL or SECAM coding, which are based on TV that uses a matrix transform called YUV.

2. 1 YIQ Color Model

YIQ is used in NTSC color TV broadcasting. In this model, the transmitted color signals are formed by encoding RGB picture information into two parts, the Luminance Y (brightness information) and the Chrominance (color information). The color Hue information is given by the chrominance (phase) angle Q and the Saturation information is given by the chrominance amplitude I. it is clear that gray pixels generate zero (I,Q) chrominance signals. Hence, color TV's can be displayed on black-and-white televisions by just using the Y signal.

The transformation of the RGB signals to YIQ color space is given in the below transformation matrix . This equation is a simple representation of the NTSC encoding scheme using in American broadcasting televisions.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & 0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

We can go backward from (YIQ) to (RGB) by ***inverting*** the transformation matrix above.

2.2 YUV Color Model

Similar to YIQ representation, but for European TV transmission system, initially YUV coding was used for PAL and SECAM analogue video which is used in western Europe. The transformation RGB signals to YUV color space is given by the below transformation matrix.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

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 (2). 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.

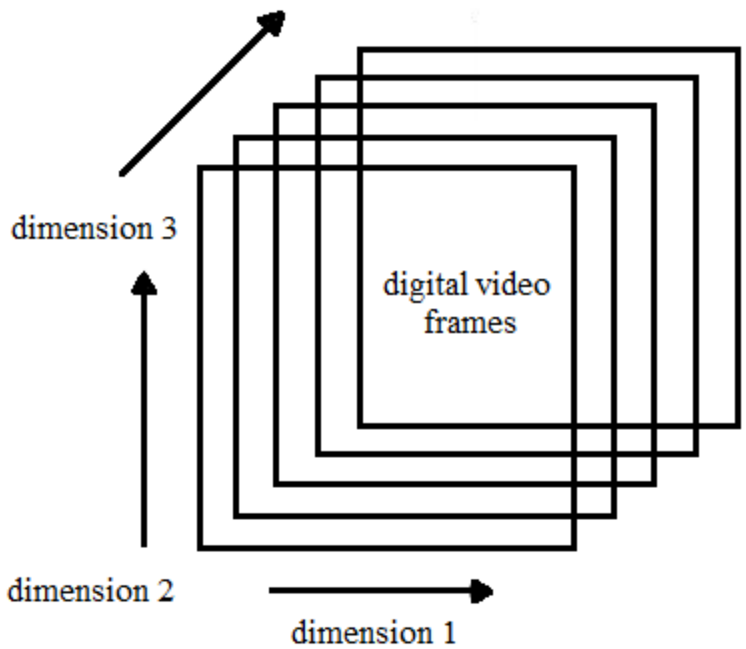


Figure (1) : the dimensionality of video

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 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.

<u>Ex</u>

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* .

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 .

Many applications like surveillance systems , Video teleconferencing can be improved by using the panoramic video concept.

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. Type of Video Signals

Video signals can be organized in three different ways: Component video, Composite video, and S-video. In earlier production, digital video was in the form of composite video. Modern digital video generally uses component video, although RGB signals are first converted into a certain type of color opponent space, such as YUV, YIQ.

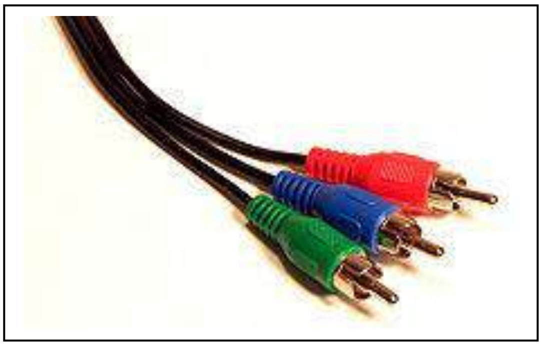
4.1 Component Video (Used in Modern Production)

This type of video signals make use of three separate video signals for kind Red, Green, and Blue image planes . This is referred to as component video. This type of system has three wires (and connectors) connecting the camera or other device to a TV or monitor. this type of video signals sends the three colors in three different signals ***so it***

requires high bandwidth and good synchronization between the three signals, this type require three cables to transmit each color signal alone.

Color signals are not restricted to always being RGB separation, as we saw in color model for video, we can form three signals via a luminance (intensity) chrominance (color) transformation of the RGB signals- for example YIQ or YUV. In contrast, most computer systems use component video , with separate signals for R, G, and B signals.

For any color separation scheme, component video gives the ***best*** color representation, since there is no "***crosstalk***" between three different channels, Unlike composite video or S-video. Component video , however requires ***more bandwidth*** and ***good synchronization*** of the three component.



4.2 Composite Video (Used in Earlier Production)

In composite video , Color (chrominance) and Intensity (luminance) signals are ***mixed*** into ***single*** carrier wave. Chrominance is a composite of two color component (I and Q , or U and V). this is the type of signal used by broadcast color TV's, its downward compatible with black-and-white TV.

In NTSC TV, for example, I and Q are combined into chroma signal, and a color subcarrier then puts the chroma signal at the higher frequency end of the channel shared with the luminance signal. The chrominance and luminance component can be ***separated*** at the ***receiver end*** , and the two color component can be further recovered.

When connecting to TV's, composite video uses ***only one wire*** (and hence one connecter, such as a BNC connecter at each end of the coaxial cable), and video color signals are mixed , not send separately . the audio signal is another addition to this one signal. Since color information is mixed and both color and intensity are wrapped into the same signal, some ***interference*** between the luminance and chrominance signal is ***inevitable***.

from left to right: Ch-2 (right) audio,

Ch-1 (left) audio, composite video.

4.3 S-Video

As a compromise , S-video (separated video , or super video) uses two wires : one for luminance and the other for the composite chrominance signals. As a result , there is ***less crosstalk*** between the color information and the crucial gray scale information.

The reason for placing luminance into its own part of the signal is that black-and-white information is ***crucial for visual perception***. As noted, humans are able to differentiate spatial resolution in grayscale images much better than for color part of color images (as opposed to the "black-and-white" part). Therefore, color information sent can be much less accurate than intensity information. We can see only fairly large blobs of color, so it makes sense to send less color detail.

From top to bottom: IEEE 1394 connector for digital video, LANC connector for camera control, TRRS connector for composite video and 2 channels of audio, S-video connector.

4.4 HDMI (High Definition Multimedia Interface)

HDMI is Compact audio/video interface. It can Carry both uncompressed high definition video along with all existing muti-channel audio formats. It can be Used in HD DVD Players, Digital Cameras and camcorders, Personal Computers, Tablets, Mobile phones, Projector.

HDMI and Component Video both are video standards which support a variety of resolutions, but which deliver the signal from the source to the display in very different ways. The principal important difference is that an HDMI cable delivers the signal in a ***digital format***, much the same way that a file is delivered from one computer to another along a network, while Component Video is an ***analog format***, delivering the signal not as a bit stream, but as a set of continuously varying voltages representing (albeit indirectly, as we'll get to in a moment) the red, green and blue components of the signal.

Both HDMI cable and Component Video cable deliver signals as three discrete color components, together with sync information which allows the display to determine when a new line, or a new frame, begins.

5 . Analog Video

Most TV is still send and received as an ***analog signal***. Once the electrical signals is 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, this 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***.

In fact, the odd lines (starting from 1) and up at the middle of a line at the end of the odd field, and the even scan starts at the half-way point. Figure (2) shows the scheme used. First the solid(odd) lines are traced- P to Q, then R to S , and so on, ending at T- then the even field start at U and end at V. the scan lines are ***not horizontal*** because a ***small voltage*** is applied, moving the electron beam ***down overtime***.

Interlacing was invented because, when standard were being defined, ***it was difficult to transmit the amount of information in a frame quickly enough to avoid flicker***. The double number of fields presented to the eye ***reduces perceived flicker***.

Because of interlacing, the odd and even lines are displaced in time from each other. This is generally not noticeable except when fast action is taking place onscreen, when blurring may occur.

The ***only problem*** could appear in interlacing is ***blurrin***g effect, when there is a ***fast motion*** in the video scene because each frame is divided into two frames.

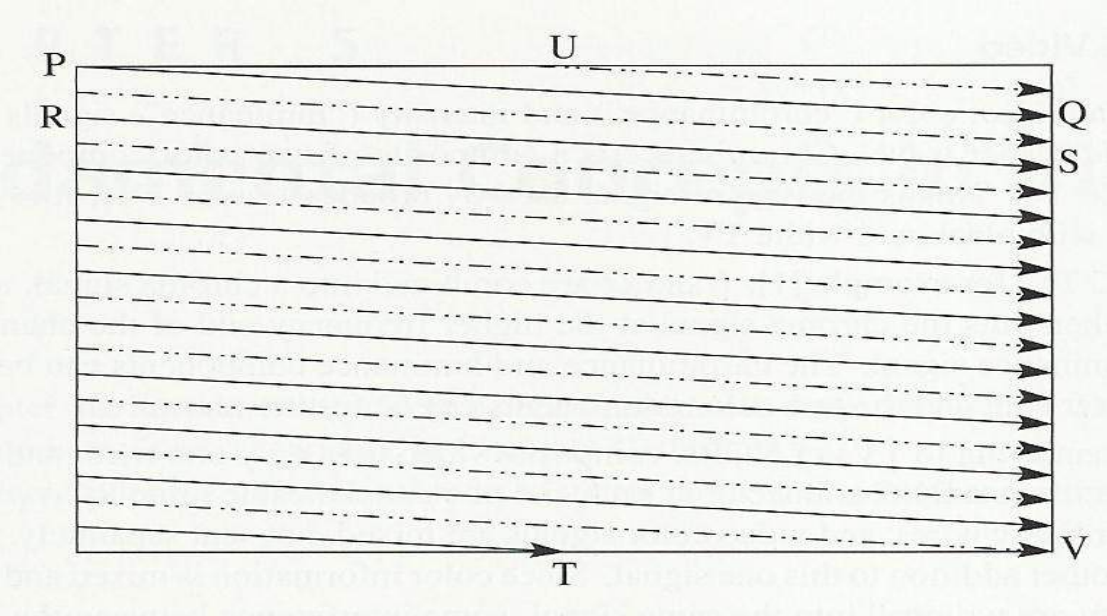


Figure (2): interlaced Raster Scan

5.1 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).

5.2 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**.

3.3 SECAM video

SECAM , which was invented by the French, is the third major broadcast TV standard. SECAM stand for **S**ystem **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.

Ex

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 emit one frame.

Sol

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

7. Digitizing Video

Digital video is more recent than analog video, digital video file is represented as series of bits and stored on some medium like hard disk, CD or any other medium.

Digital video captured by cameras that will store it on disk or connected to a computer and store the video directly on it.

Analog videos can be converted to digital form by a process called digitization, ***digitization*** converts continues analog signal to series of samples (sampling) then each sample rounded to a fixed set of numbers(quantization) as shown in Figure 3.

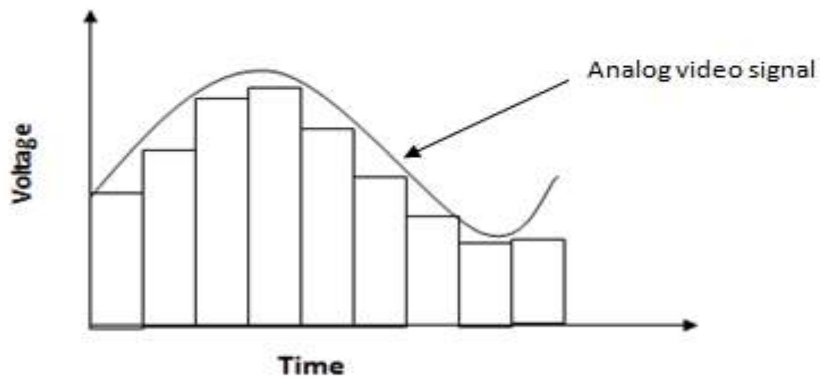


Figure 3 : digitizing video

By digitizing (sampling and quantization) an analog video the result will be the digital representation of the video .

A video consists of a sequence of frames. If the frames are displayed on the screen fast enough, we get an impression of motion. The reason is that our eyes cannot distinguish the rapidly flashing frames as individual ones. There is no standard number of frames per second; 25 frames per second is common. However, to avoid a condition known as flickering, a frame needs to be refreshed. The TV industry repaints each frame twice. This means 50 frames need to be sent, or if there is memory at the sender site, 25 frames with each frame repainted from the memory.

Each frame is divided into small grids, called picture elements or pixels. For black and white TV, each 8-bit pixel represents one of 256 different gray levels. For a color TV, each pixel is 24 bits, with 8 bits for each primary color (red, green, and blue).

We can calculate the number of bits in 1 s for a specific resolution. In the lowest resolution a color frame is made of 1024 × 768 pixels. This means that we need

$$2 \times 25 \times 1024 \times 768 \times 24 = 944 \text{ Mbps}$$

This data rate needs a very high data rate technology such as SONET. To send video using lower-rate technologies, we need to compress the video.

8. 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.

Streaming applications and techniques are being very important and vital with the growth and expanding of networks and internet in recent years. Streaming technology is a cost effective techniques, where it save significant amount of travel and time.

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.

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 .

8.1 Video Streaming Methods

Video streaming is the transmission of large video files from the source or producer to the destination or consumer, ***the producer usually called the server and the consumer called the client***, client could be one or more, video delivery or streaming can be divided into three types diverse in properties or operating conditions or communication application, communication may be Unicast (one - to-one), Multicast (one-to-many) and Broadcast (one-to-all) .

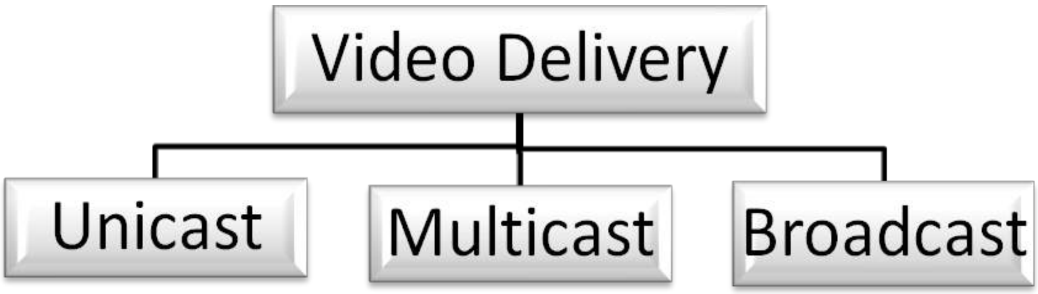


Figure 4 : Video Streaming methods

Unicast means that the connection is between a single sender and single receiver like video call and unicast video streaming over the internet from one server to one client only, the streaming usually ***one way only from the sender to the receiver no feedback*** from the receiver to the sender but sometimes there is a back channel between the sender and receiver, if this channel exists the receiver can feedback to the sender (like its statues if the client has enough free buffer or the speed of consumption of the received packets) which allow the sender to adapt with these information like slowing the sending rate.

Video ***broadcast*** means send the video or multimedia file to all receivers no exception, a transmission like this requires build a system that can ***efficiently*** deliver a video to multiple clients and it is ***able to handle*** the different specifications for each client like channel characteristics either it is a circuit switching or packet switching, client's characteristics like the client's platform speed and available buffer, the network that client is participating to characteristics usually a broadcast system designed for the worst case channel like television broadcast and ***no feedback*** which limits the ability of the system to adapt.

Multicast is between unicast and broadcast where sender sends a video to a specific group of receivers only like people grouped in one network or IP-multicast over the internet only the subscribed people are able to receive and watch the broadcast ***no one out the group allowed to watch the casting without authorization .***

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.

Videos over the internet can be classified into two types according to the time of creation, Live Videos and On Demand videos.

1. **Live video** : live streaming is when the transmission of the video is pushed to the client viewer as the connection is created. live video transmission is a ***real time*** transmission of video frames through a local area network or through the internet so that it will be seen on personal computer , smart phones or mobile devices. It is a video captured and encoded and transmitted after ***live connection*** has been established between the sender and the receiver or multi receivers at the same time live event is happening like video calls, video conferences, interactive games and live broadcast for sport events and concerts.

   The Live video pass through **many levels** during its journey from the sender to the receiver: **captured, encoded, streamed, received, decoded and viewed** all these steps implemented in the same time the live event is happening so ***timing*** with this type of videos is ***important issue*** video packets should be delivered in its specific time or it will be useless, ***delay is not an option***, quality of received video in live video delivery is not an important issue but the delivery in time and recovery from packet loss in short time is very important.

   Latency Time : in live video streaming applications "latency" is the time between capturing the frame at the server to the time of displaying the frame at the client. The amount of total delay which is also called " End-to-End Latency" is the summation of : processing time, the time required for the transmission and rendering time. ***Low latency is a vital requirement in the live streaming application".***

2. **Video on Demand** : The alternative is on Demand video or (pre encoded video), where the viewer ***request*** the video from a content server (library) . It is a video that recorded and encoded and ***stored locally*** or remotely in a ***prior time*** and available at any time a client will request, most of the time On Demand videos are used like videos on CD's or DVD's, collage lectures, medical operations, video clips and movie trailers, when client send a request to watch an on demand video or existed

video server starts immediately stream the requested video to the client no need to online processing for this type of videos.

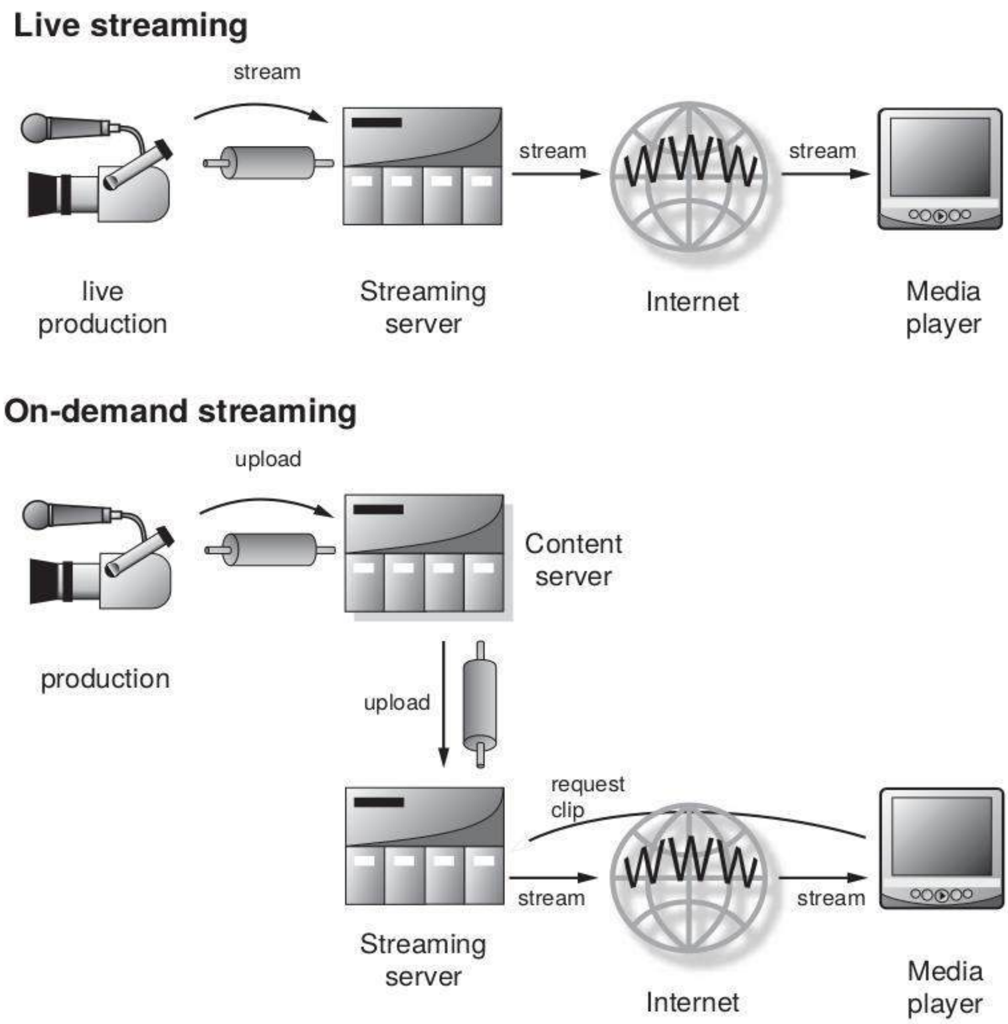


Figure 5 : live and on-demand streaming

8.2 Video Streaming Protocols

Video transmission from one place to the other is called streaming, streaming ***could be in the same station*** like copying videos from CD or DVD to the Hard Disk or from remote station to the other in this case internet connection is required.

Video ***streaming over the internet requires dedicated protocols*** to accomplish this operation, there are many streaming protocols and the most used protocols as shown in

Figure 6 For streaming live videos the Real Time Transport Protocol (RTP) and Real Time Control protocol (RTCP) over User Datagram Protocol (UDP) and Internet Protocol (IP), and there are other protocols like Hyper Text Transmission Protocol (HTTP) over Transmission control Protocol (TCP) and Internet Protocol (IP) more suitable for transmitting on demand videos over the internet.

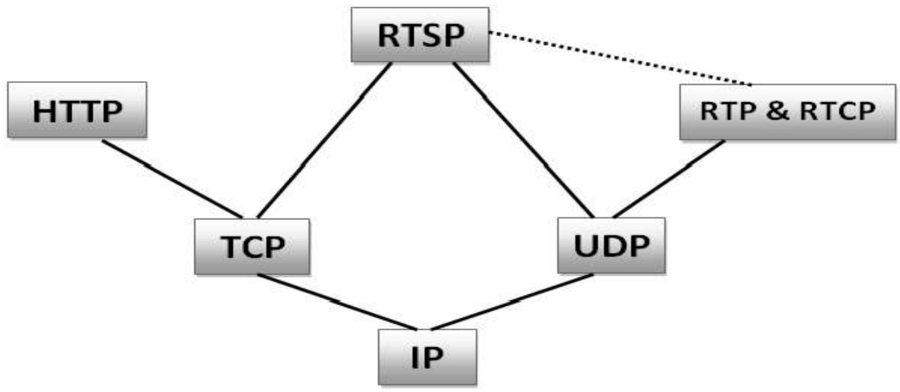


Figure 6 : Streaming Protocols

9. 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***.

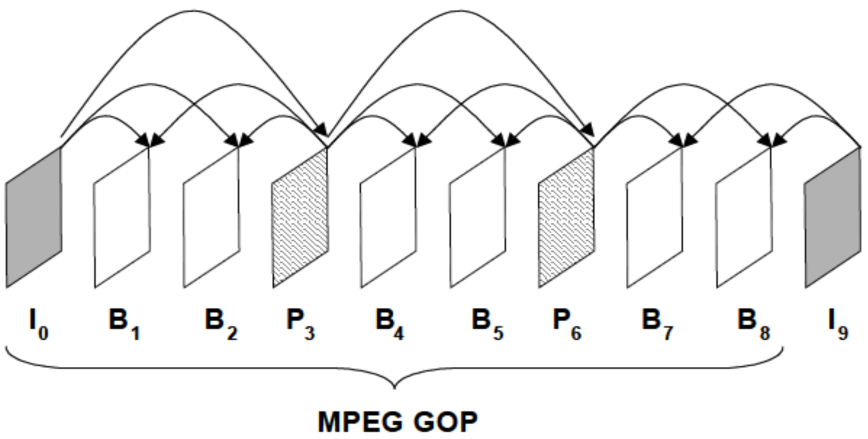


Figure 7: MPEG group of images [ALT02]

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.

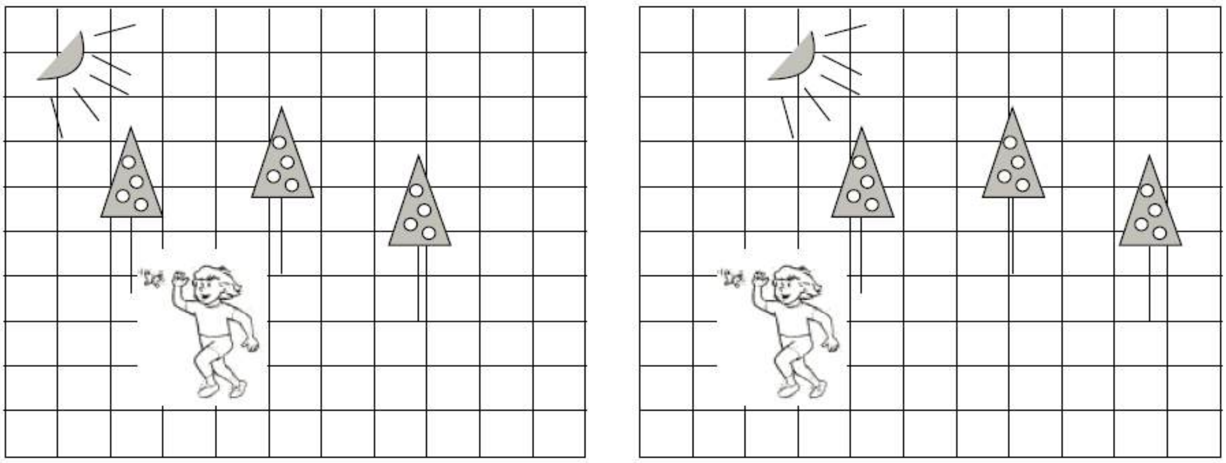
10 . motion estimation

The references between the different types of frames are realized by a process called motion estimation or **motion compensation**. The correlation between two frames in terms of **motion** is represented by a **motion vector**. The resulting frame correlation, and therefore the pixel arithmetic difference, strongly depends on how good the motion estimation algorithm is implemented. **Good estimation** results in **higher compression ratios** and **better quality** of the coded video sequence. However, motion estimation is computational intensive operation, which is often not well suited for real time applications.

The motion estimation paramerers

- Frame Segmentation - The Actual frame is divided into non-overlapping blocks (macro blocks) usually 8x8 or 16x16 pixels. ***The smaller the block sizes are chosen, the more vectors need to be calculated;*** the block size therefore is a critical factor in terms of time performance, but also in terms of quality: if the blocks are **too large**, the motion matching is most likely **less correlated**. If the blocks are **too small**, it is probably, that the algorithm will try to **match noise**. MPEG uses usually block sizes of 16x16 pixels.

- Search Threshold - In order to minimize the number of expensive motion estimation calculations, they are only calculated if the difference between two blocks at the same position is higher than a threshold, otherwise the whole block is transmitted.
- Block Matching - In general block matching tries, to "stitch together" an actual predicted frame by using snippets (blocks) from previous frames. The process of block matching is the most time consuming one during encoding. In order to find a matching block, each block of the current frame is compared with a past frame within a search area. Only the luminance information is used to compare the blocks, but obviously the color information will be included in the encoding. The search area is a critical factor for the quality of the matching. It is more likely that the algorithm finds a matching block, if it searches a larger area. Obviously the number of search operations increases quadratically, when extending the search area. Therefore too large search areas slow down the encoding process dramatically. To reduce these problems often rectangular search areas are used, which take into account, that horizontal movements are more likely than vertical ones.



(a) (b)

Forward prediction in which the Reference frame is taken to be a previous frame