Computer Graphics & Multimedia Application

M.Sc (CA) 202

SELF LEARNING MATERIAL

DIRECTORATE

OF DISTANCE EDUCATION

SWAMI VIVEKANAND SUBHARTI UNIVERSITY

MEERUT – 250 005,

UTTAR PRADESH (INDIA)

SLM Module Developed By :

Author:

Reviewed by :

Assessed by:

Study Material Assessment Committee, as per the SVSU ordinance No. VI (2)

Copyright © **Gayatri Sales**

DISCLAIMER

No part of this publication which is material protected by this copyright notice may be reproduced or transmitted or utilized or stored in any form or by any means now known or hereinafter invented, electronic, digital or mechanical, including photocopying, scanning, recording or by any information storage or retrieval system, without prior permission from the publisher.

Information contained in this book has been published by Directorate of Distance Education and has been obtained by its authors from sources be lived to be reliable and are correct to the best of their knowledge. However, the publisher and its author shall in no event be liable for any errors, omissions or damages arising out of use of this information and specially disclaim and implied warranties or merchantability or fitness for any particular use.

Published by: Gayatri Sales

Typeset at: Micron Computers **Printed at:** Gayatri Sales, Meerut.

COMPUTER GRAPHICS & MULTIMEDIA APPLICATIONS

UNIT-I

Introduction: The Advantages of Interactive Graphics, Representative Uses of Computer Graphics, Classification of Application Development of Hardware and software for computer Graphics, Conceptual Framework for Interactive Graphics, Overview, Scan: Converting Lines, Scan Converting Circles, Scan Converting Ellipses.

UNIT-II

Hardcopy Technologies, Display Technologies, Raster-Scan Display System, Video Controller, Random-Scan Display processor, Input Devices for Operator Interaction, Image Scanners, Working exposure on graphics tools like Dream Weaver, 3D Effects etc.

Clipping

Southland- Cohen Algorithm, Cyrus-Beck Algorithm, Midpoint Subdivision Algorithm

UNIT-III

Geometrical Transformation

2D Transformation, Homogeneous Coordinates and Matrix Representation of 2D Transformations, composition of 2D Transformations, the Window-to-Viewport Transformations, Introduction to 3D Transformations Matrix.

UNIT-IV

Representing Curves & Surfaces

Polygon meshes parametric, Cubic Curves, Quadric Surface;

Solid Modeling

Representing Solids, Regularized Boolean Set Operation primitive Instancing Sweep Representations, Boundary Representations, Spatial Partitioning Representations, Constructive Solid Geometry Comparison of Representations.

UNIT-V

Introductory Concepts: Multimedia Definition, CD-ROM and the multimedia highway, Computer Animation (Design, types of animation, using different functions) Uses of Multimedia, Introduction to making multimedia – The stage of Project, hardware & software requirements to make good multimedia skills and Training opportunities in Multimedia Motivation for Multimedia usage

UNIT-I

Introduction:

The Advantages of Interactive Graphics

Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. In Many design, implementation, and construction processes today, the information pictures can give is virtually indispensable. Scientific visualization became an important field in the late 1980s, when scientists and engineers realized that they could not interpret the data and prodigious quantities of data produced in supercomputer runs without summarizing the data and highlighting trends and phenomena in various kinds of graphical representations.

Creating and reproducing pictures, however, presented technical problems that stood in the way of their widespread use. Thus, the ancient Chinese proverb "a picture is worth ten thousand words" became a cliché in our society only after the advent of inexpensive and simple technology for producing pictures---first the printing press, then Photography.

Interactive computer graphics is the most important means of producing pictures since the invention of photography and television; it has the added advantage that, with the computer, we can make pictures not only of concrete, "real-world" objects but also of abstract, synthetic objects, such as mathematical surfaces in 4D and of data that have no inherent geometry, such as survey results. Furthermore, we are not confined to static images. Although static pictures are a good means of communicating information, dynamically varying pictures are frequently even better--to time-varying phenomena, both real (e.g., growth trends, such as nuclear energy use in the United States or population movement form cities to suburbs and back to the cities). Thus, a movie can show changes over time more graphically than can a sequence of slides. Thus, a sequence of frames displayed on a screen at more than 15 frames per second can convey smooth motion or changing form better than can a jerky sequence, with several seconds between individual frames. The use of dynamics is especially effective when the user can control the animation by adjusting the speed, the portion of the total scene in view, the amount of detail shown, the geometric relationship of the objects in the another, and so on. Much of interactive graphics technology therefore contains hardware and software for user-controlled motion dynamics and update dynamics.

With motion dynamics, objects can be moved and tumbled with respect to a stationary observer. The objects can also remain stationary and the viewer can move around them

, pan to select the portion in view, and zoom in or out for more or less detail, as though looking through the viewfinder of a rapidly moving video camera. In many cases, both the objects and the camera are moving. A typical example is the flight simulator, which combines a mechanical platform supporting a mock cockpit with display screens for windows. Computers control platform motion, gauges, and the simulated world of both stationary and moving objects through which the pilot navigates. These multimillion-dollar systems train pilots by letting the pilots maneuver a simulated craft over a simulated 3D landscape and around simulated vehicles. Much simpler fight simulators are among the most popular games on personal computers and workstations. Amusement parks also offer "motion-simulator" rides through simulated terrestrial and extraterrestrial landscapes. Video arcades offer graphics-based dexterity games and racecar-driving simulators, Video Games exploiting interactive motion dynamics: The player can change speed and direction with the "gas pedal" and "steering wheel," as trees, buildings, and other cars go whizzing by. Similarly, motion dynamics lets the user fly around the through buildings, molecules, and 3D or 4D mathematical space. In another type of motion dynamics, the "camera" is held fixed, and the objects in the scene are moved relative to it. For example, a complex mechanical linkage, such as the linkage on a stream engine, can be animated by moving or rotating all the pieces appropriately.

Update dynamics is the actual change of the shape, color, or other properties of the objects being viewed. For instance, a system can display the deformations of an airplane structure in flight or the state changes in a block diagram of a nuclear reactor in response to the operator's manipulation of graphical representations of the many control mechanisms. The smoother the change, the more realistic and meaningful the result. Dynamic interactive graphics offers a large number of user-controllable modes with which to encode and communicate information: the 2D or 3D shape of objects in a picture, their gray scale or color, and the time variations of these properties. With the recent development of digital signal processing (DSP) and audio synthesis chips, audio feedback can now be provided to augment the graphical feedback and to make the simulated environment even more realistic.

Interactive computer graphics thus permits extensive, high-bandwidth user-computer interaction. This significantly enhances our ability to understand data, to perceive trends, and to visualize real or imaginary objects--indeed, to create "virtual worlds" that we can explore from arbitrary points of view. By making communication more efficient, graphics make possible higher-quality and more precise results or products, greater productivity, and lower analysis and design costs.

Representative Uses of Computer Graphics

The use of computer graphics is wide spread. It is used in various areas such as industry, business, government organizations, education, entertainment and most recently the home. Let us discuss representative uses of computer graphics in brief.

User friendliness is one of the main factors underlying the success and popularity of any system. It is now a well established fact that graphical interfaces provide in attractive and easy interaction between users and computers. The built-in graphics provided with user interfaces use visual control items such as buttons, menus, icons, scroll bar etc, which allows user to interact with computer only by mouse-click. Typing is necessary only to input text to be stored and manipulated.

In industry, business, government and educational organizations, computer graphics is most commonly used to create 2D and 3D graphics of mathematical, physical and economic functions in form of histograms, bars, and pie-chats. These graphs and charts are very useful for decision making.

The desktop publishing on personal computers allow the use of graphics for the creation and dissemination of information. Many organizations does the in-house creation and dissemination of documents. The desktop publishing allows user to create documents which contain text, tables, graphs, and other forms of drawn or scanned images or pictures. This is one approach towards the office automation.

The computer-aided drafting uses graphics to design components and systems electrical, mechanical, electromechanical and electronic devices such as automobile bodies, structures of building, airplane, slips, very large-scale integrated chips, optical systems and computer networks.
Use of graphics in simulation makes mathematic models and mechanical systems more realistic and easy to study. The interactive graphics supported by animation software proved their use in production of animated movies and cartoons films.
There is lot of development in the tools provided by computer graphics. This allows user to create artistic pictures which express messages and attract attentions. Such pictures are very useful in advertising.
By the use of computer now it is possible to control various processes in the industry from a remote control room. In such cases, process systems and processing parameters are shown on the computer with graphic symbols and identification. This makes it easy for operator to monitor and control various processing parameters at a time.

Classification of Application Development of Hardware and software for computer Graphics

a. Output Technology:

Output is the term denoting either an exit or changes which exit a system and which activate/modify a process. It is an abstract concept, used in the modeling, system(s) design and system(s) exploitation.

Pictures created and manipulated through the use of computer devices. The term computer graphics generally pertains to any computer device or program that makes a computer capable of displaying and manipulating pictures. For example: a laser printer is said to be a computer graphics device because it allows the computer to output pictures; likewise, a computer display monitor can display pictures. Computer graphics are used for various applications including publishing, education, entertainment, and advertising, or wherever pictures are deemed reasonable or necessary in the creation of a message. They are also used very effectively in situations where there is a need for computed data to be visualized, such as in statistical charts or graphs of mathematical data. Most computer graphics can also be drawn by an artist, but the computer can accomplish much more in a much shorter period of time. One of the major benefits of computer graphics is that images can be manipulated with relative ease and that a multitude of visual effects are possible because the images can be played with over and over again until a desired effect is achieved.

A general purpose computer has four main components: the arithmetic logic unit (ALU), the control unit, the memory, and the input and output devices (collectively termed I/O). These parts are interconnected by busses, often made of groups of wires.

Inside each of these parts are thousands to trillions of small electrical circuits which can be turned off or on by means of an electronic switch. Each circuit represents a bit (binary digit) of information so that when the circuit is on it represents a "1", and when off it represents a "0" (in positive logic representation). The circuits are arranged in logic gates so that one or more of the circuits may control the state of one or more of the other circuits.

The control unit, ALU, registers, and basic I/O (and often other hardware closely linked with these) are collectively known as a central processing unit (CPU). Early CPUs were composed of many separate components but since the mid-1970s CPUs have typically been constructed on a single integrated circuit called a microprocessor.

I/O is the means by which a computer exchanges information with the outside world.[27] Devices that provide input or output to the computer are called peripherals.[28] On a typical personal computer, peripherals include input devices like the keyboard and mouse, and output devices such as the display and printer. Hard disk drives, floppy disk

drives and optical disc drives serve as both input and output devices. Computer networking is another form of I/O.

b. Input Technology:

Advances in computer graphics have transformed how we use computers. Computer graphics has given us the "mouse" input device, "what-you-see-is-what-you-get" document preparation systems, the computer-aided design system used to create the Boeing 777, the ability to visualize molecular dynamics and other scientific phenomena, the animation used in educational software and the advertising and entertainment industries, and virtual reality systems whose applications range from architectural prototyping to surgical training to entertainment. Today, every user of a computer benefits from computer graphics, even in applications such as word processors, spreadsheets, databases, and project planners. Because of user-friendly graphical user interfaces, pre-schoolers now routinely use computers, a revolution undreamt of even a few years ago.

While everyone is familiar with the mouse, multiple "windows" on computer screens, and stunningly realistic images of everything from animated logos in television advertisements to NASA animations of spacecraft flying past Saturn, few people realize that these innovations were spawned by federally sponsored university research. Without far-sighted support from agencies such as the Department of Defense Advanced Research Projects Agency and the National Science Foundation, computer graphics and the multi-billion-dollar industry it makes possible would have developed much more slowly, and perhaps not predominantly in the U.S.

c. Software Technology:

The aim of this work is to show how the most advanced technology together with spatial analysis can be usefully employed to investigate historical and archaeological phenomena. In this note some preliminary results are shown. Two geographical information systems (GIS) were structured in an integrated way. The first GIS is a vector-like system while the other is a raster-like one. Moreover, some applications regarding the environmental reconstruction of a part of the investigated area are proposed. Then the identification and the modeling of archaeological site maps by means of point pattern analysis are proposed. Finally, an auto-logistic model to predict archaeological site is presented. This topic is currently under investigation

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software. The term software engineering first appeared in the 1968 NATO Software Engineering Conference and was meant to provoke thought regarding the current "software crisis" at the time. Since then, it has continued as a profession and field of study dedicated to creating software that is of higher quality, more affordable, maintainable, and quicker to build. Since the field is still relatively young compared to its sister fields of engineering, there is still much debate around what software engineering actually is, and if it conforms to the classical definition of engineering. It has grown organically out of the limitations of viewing software as just programming. Software development, a much used and more generic term, does not necessarily subsume the engineering paradigm

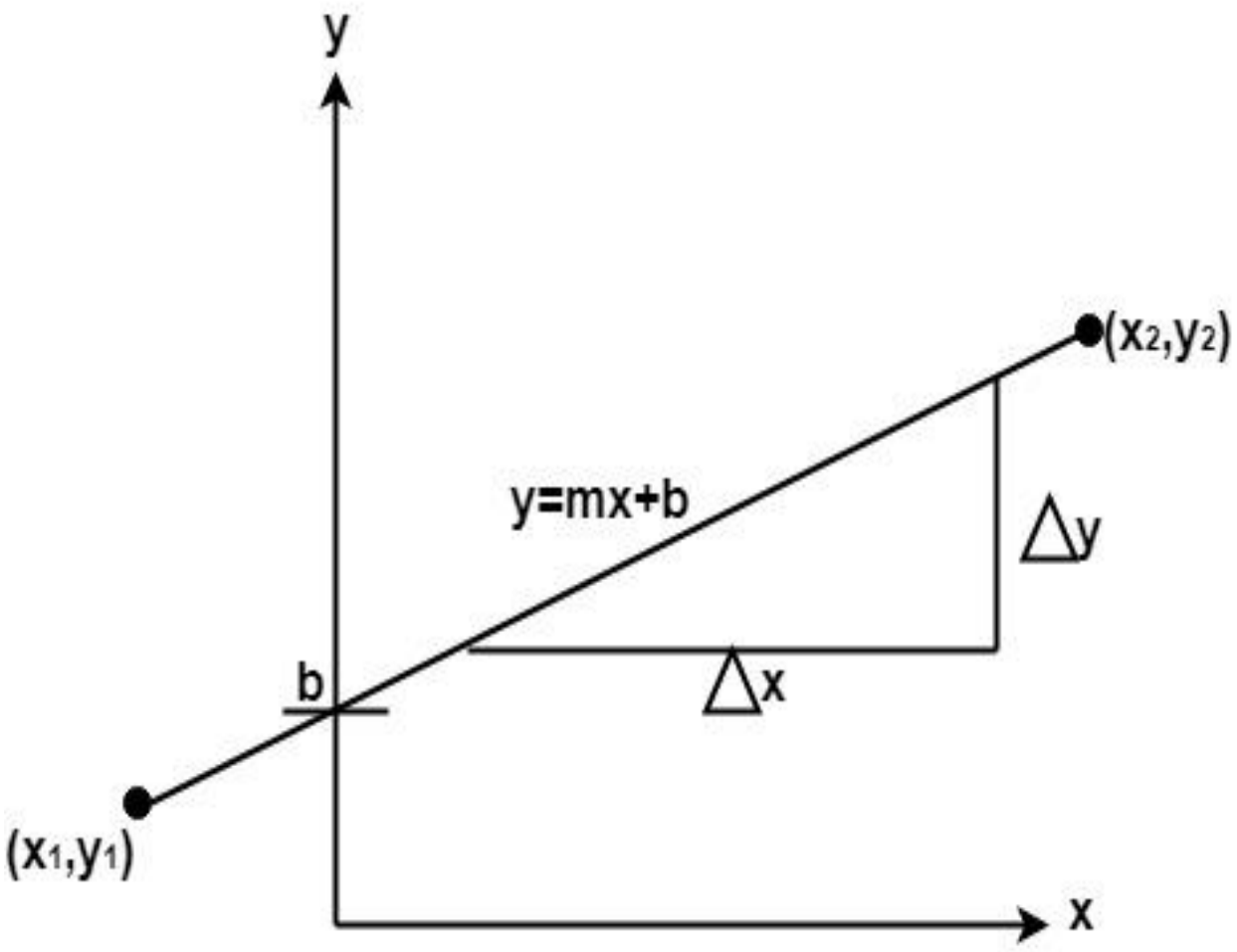
Conceptual Framework for Interactive Graphics

Conceptual Framework has the following elements:

- Graphics Library - Between application and display hardware there is graphics library / API.
- Application Program - An application program maps all application objects to images by invoking graphics.
- Graphics System – An interface that interacts between Graphics library and Hardware.
- Modifications to images are the result of user interaction.

Scan:
Converting Lines

A straight line may be defined by two endpoints & an equation. In fig the two endpoints are described by (x_1,y_1) and (x_2,y_2). The equation of the line is used to determine the x, y coordinates of all the points that lie between these two endpoints.



Using the equation of a straight line, $y = mx + b$ where $m = \frac{\Delta y}{\Delta x}$ & b = the y interrupt, we can find values of y by incrementing x from $x = x_1$, to $x = x_2$. By scan-converting these calculated x, y values, we represent the line as a sequence of pixels.

Properties of Good Line Drawing Algorithm:

1. Line should appear Straight:

We must appropriate the line by choosing addressable points close to it. If we choose well, the line will appear straight, if not, we shall produce crossed lines.

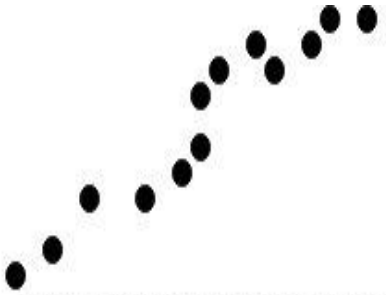


Fig: O/P from a poor line generating algorithm

The lines must be generated parallel or at 45° to the x and y-axes. Other lines cause a problem: a line segment through it starts and finishes at addressable points, may happen to pass through no another addressable points in between.

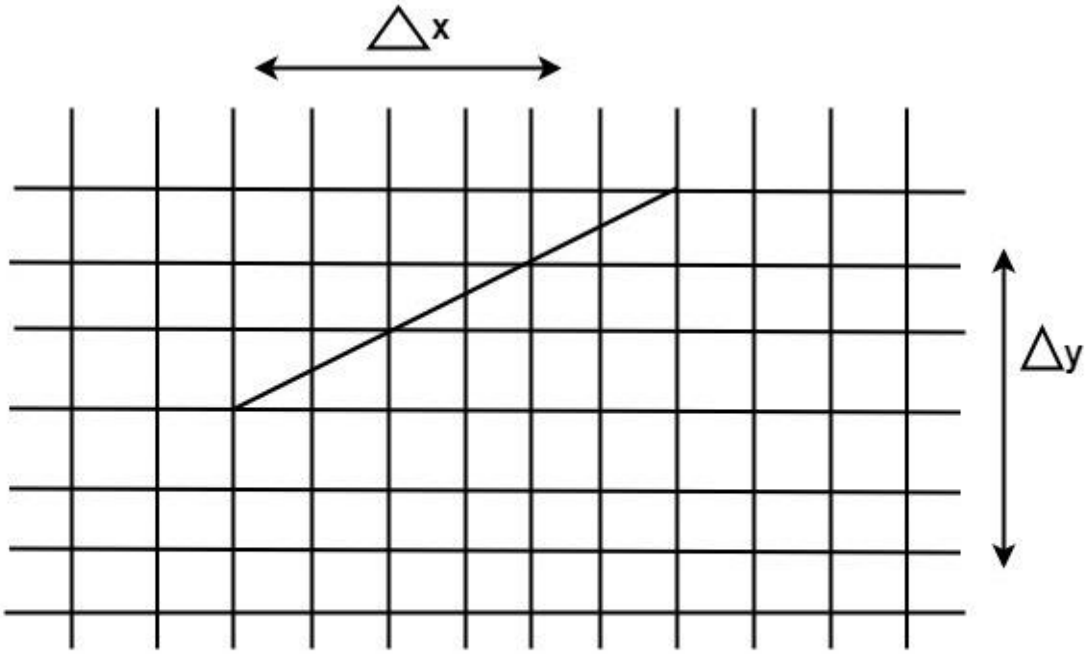


Fig: A straight line segment connecting 2 grid intersection may fail to pass through any other grid intersections.

2. Lines should terminate accurately: Unless lines are plotted accurately, they may terminate at the wrong place.

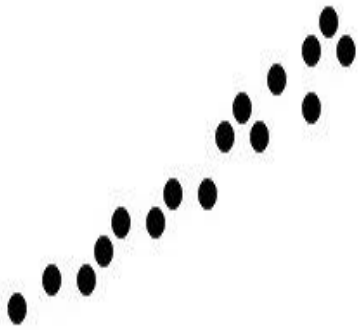


Fig: Uneven line density caused by bunching of dots.

3. Lines should have constant density: Line density is proportional to the no. of dots displayed divided by the length of the line.

To maintain constant density, dots should be equally spaced.

4. Line density should be independent of line length and angle: This can be done by computing an approximating line-length estimate and to use a line-generation algorithm that keeps line density constant to within the accuracy of this estimate.

5. Line should be drawn rapidly: This computation should be performed by special-purpose hardware.

Algorithm for line Drawing:

1. Direct use of line equation
2. DDA (Digital Differential Analyzer)
3. Bresenham's Algorithm

Direct use of line equation:

It is the simplest form of conversion. First of all scan P_1 and P_2 points. P_1 has co-ordinates (x_1',y_1') and (x_2' y_2').

Then m = (y_2',y_1')/(x_2',x_1') and b = $y_1^1 + mx_1^1$

If value of |m|≤1 for each integer value of x. But do not consider x_1^1 and x_2^2

If value of |m|>1 for each integer value of y. But do not consider y_1^1 and y_2^2

Example: A line with starting point as (0, 0) and ending point (6, 18) is given. Calculate value of intermediate points and slope of line.

Solution: P_1 (0,0) P_7 (6,18)

x_1=0
y_1=0
x_2=6
y_2=18

$$M = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{18-0}{6-0} = \frac{18}{6} = 3$$

We know equation of line is

y =m x + b
y = 3x + b.....equation (1)

put value of x from initial point in equation (1), i.e., (0, 0) x =0, y=0

0 = 3 x 0 + b
0 = b ⟹ b=0

put b = 0 in equation (1)

y = 3x + 0
y = 3x

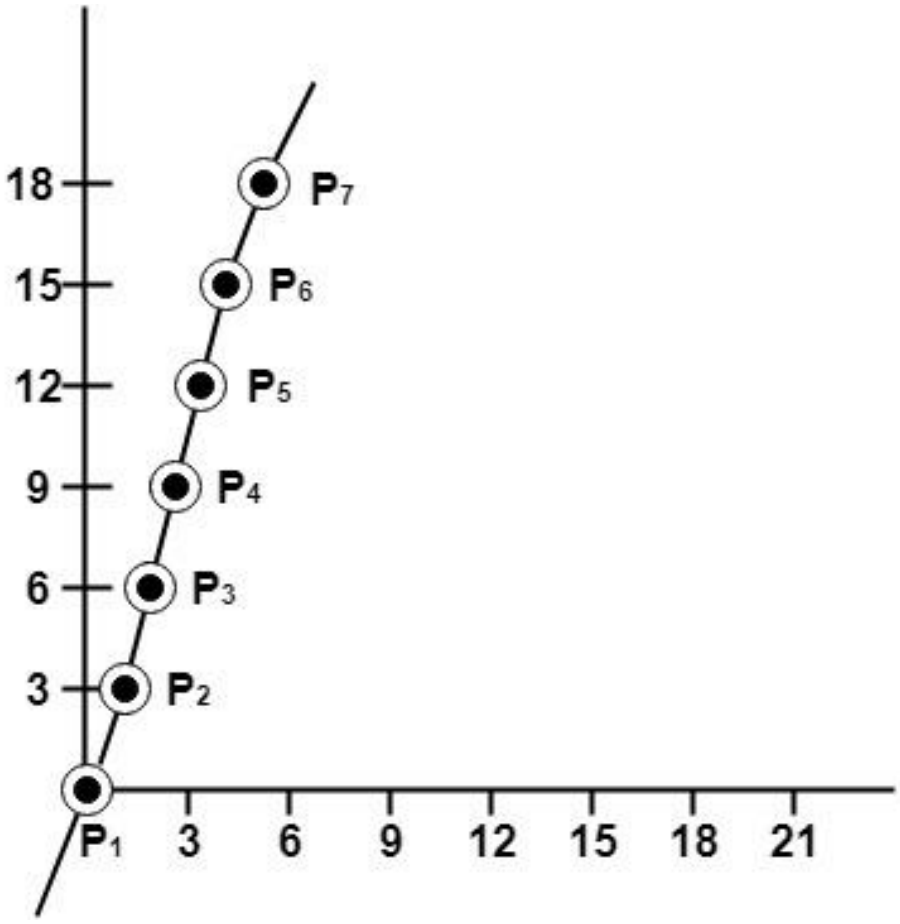
Now calculate intermediate points

Let x = 1 ⟹ y = 3 x 1 ⟹ y = 3
Let x = 2 ⟹ y = 3 x 2 ⟹ y = 6
Let x = 3 ⟹ y = 3 x 3 ⟹ y = 9
Let x = 4 ⟹ y = 3 x 4 ⟹ y = 12
Let x = 5 ⟹ y = 3 x 5 ⟹ y = 15
Let x = 6 ⟹ y = 3 x 6 ⟹ y = 18

So points are P_1 (0,0)

P_2 (1,3)
P_3 (2,6)
P_4 (3,9)
P_5 (4,12)

P_6 (5,15)
P_7 (6,18)



Algorithm for drawing line using equation:

Step1: Start Algorithm

Step2: Declare variables x_1,x_2,y_1,y_2,dx,dy,m,b,

Step3: Enter values of x_1,x_2,y_1,y_2.
The (x_1,y_1) are co-ordinates of a starting point of the line.
The (x_2,y_2) are co-ordinates of a ending point of the line.

Step4: Calculate dx = x_2- x_1

Step5: Calculate dy = y_2-y_1

Step6: Calculate m = $\frac{dy}{dx}$

Step7: Calculate b = y_1-m* x_1

Step8: Set (x, y) equal to starting point, i.e., lowest point and x_{end}equal to largest value of x.

If dx < 0
then x = x_2
y = y_2
x_{end}= x_1
If dx > 0
then x = x_1
y = y_1
x_{end}= x_2

Step9: Check whether the complete line has been drawn if x=x_{end}, stop

Step10: Plot a point at current (x, y) coordinates

Step11: Increment value of x, i.e., x = x+1

Step12: Compute next value of y from equation y = mx + b

Step13: Go to Step9.

Program to draw a line using Line Slope Method

```
#include <graphics.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

class bresen
{
float x, y, x1, y1, x2, y2, dx, dy, m, c, xend;
public:
void get ();
void cal ();
};
void main ()
```

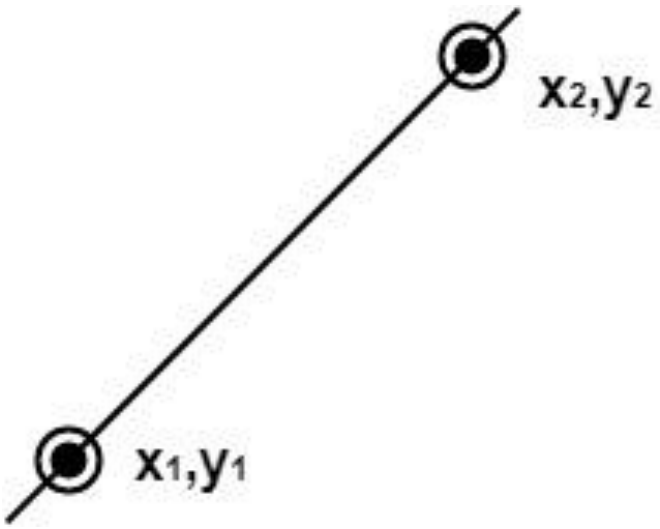
```
{
bresen b;
b.get ();
b.cal ();
getch ();
}
Void bresen :: get ()
{
print ("Enter start & end points");
print ("enter x1, y1, x2, y2");
scanf ("%f%f%f%f",sx1, sx2, sx3, sx4)
}
void bresen ::cal ()
{
/* request auto detection */
int gdriver = DETECT,gmode, errorcode;
/* initialize graphics and local variables */
initgraph (&gdriver, &gmode, " ");
/* read result of initialization */
errorcode = graphresult ();
if (errorcode ! = grOK)    /*an error occurred */
{
printf("Graphics error: %s \n", grapherrormsg (errorcode);
printf ("Press any key to halt:");
getch ();
exit (1); /* terminate with an error code */
}
dx = x2-x1;
dy=y2-2y1;
m = dy/dx;
c = y1 - (m * x1);
if (dx<0)
{
x=x2;
```

```
y=y2;
xend=x1;
}
else
{
x=x1;
y=y1;
xend=x2;
}
while (x<=xend)
{
putpixel (x, y, RED);
y++;
y=(x*x) +c;
}
}
```

OUTPUT:

Enter Starting and End Points

Enter (X1, Y1, X2, Y2) 200 100 300 200



Scan Converting Circles

A circle is an eight-way symmetric shape. All quadrants of a circle are the same. There are two octants in each quadrant of a circle. If we know the value of any point, then we can easily calculate the values of the remaining seven points by using the eight-way symmetry method. A circle is a shape consist of a line called the circumference. All the straight lines drawn from a particular point within the circle to the circumference are always equal. A circle is a round shape that has no corner or sides.
"A circle can be defined as a combination of points that all points are at the same distance (radius) from the center point." We can express a circle by the following equation-

$$(P - P_c)^2 + (Q - Q_c)^2 = r^2$$

The above equation can be written as-

$$(P)^2 + (Q)^2 = r^2 \qquad \{r = \text{radius}\}$$

If we want to draw a circle, then we will consider it by its origin point. Let us assume a point $P_1(R, S)$ then we can represent the other seven points as follows-

$P_2(R, -S)$

$P_3(-R, -S)$

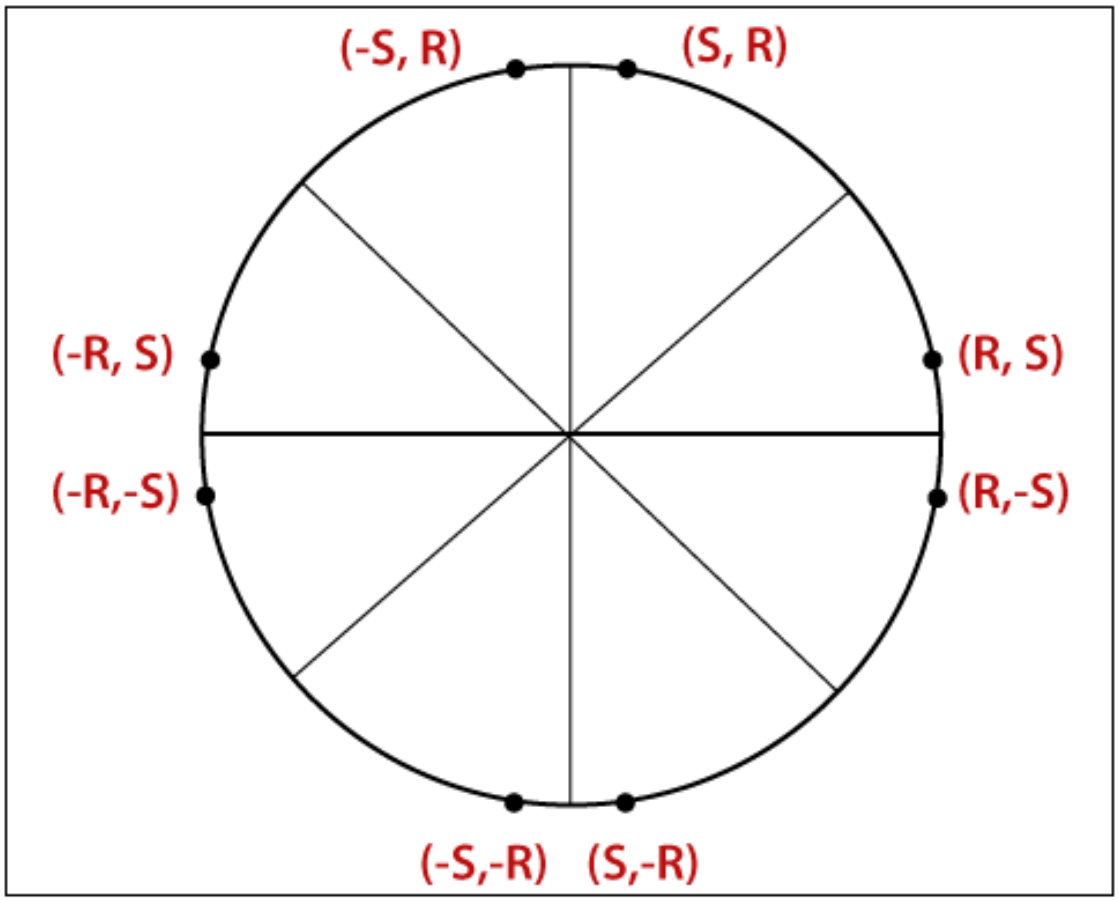
$P_4(-R, S)$

$P_5(S, R)$

$P_6(S, -R)$

$P_7(-S, -R)$

$P_8(-S, R)$



We can also represent eight points of the circle on the computer screen by using of put pixel function ().

Putpixel (R, S, Color)

Putpixel (R, -S, Color)

Putpixel (-R, -S, Color)

Putpixel (-R, S, Color)

Putpixel (R, S, Color)

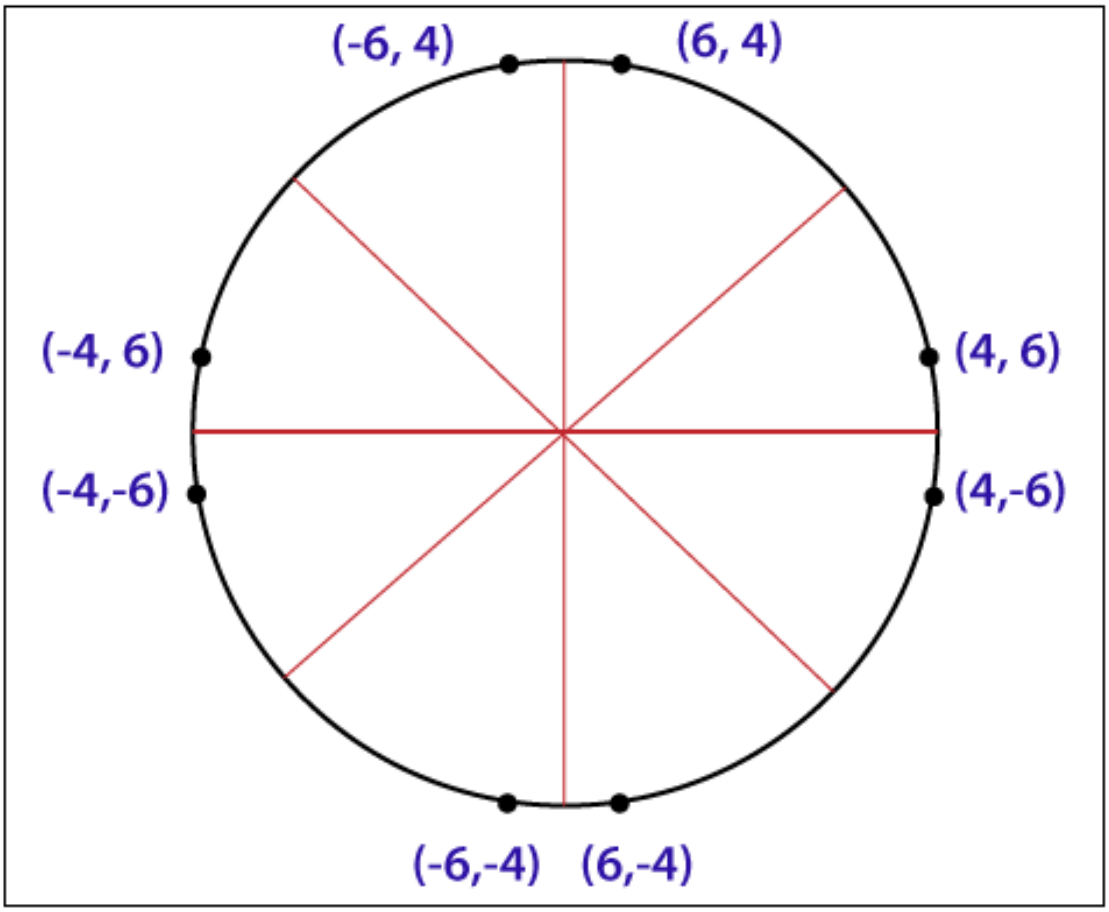
Putpixel (R, -S, Color)

Putpixel (-R, -S, Color)

Putpixel (-R, S, Color)

Example: Let, we have taken a point (4, 6) of a circle. We will calculate the remaining seven points by using of reflection method as follows-

The seven points are – (4, -6), (-4, -6), (-4, 6), (4, 6), (4, -6), (-4, -6), (-4, 6)



There are two following standard methods to define a circle mathematically.

- A circle with a second-order polynomial equation.
- A circle with trigonometric/ polar coordinates.

A circle with the second-order polynomial equation–

$$y^2 = r^2 - x^2$$

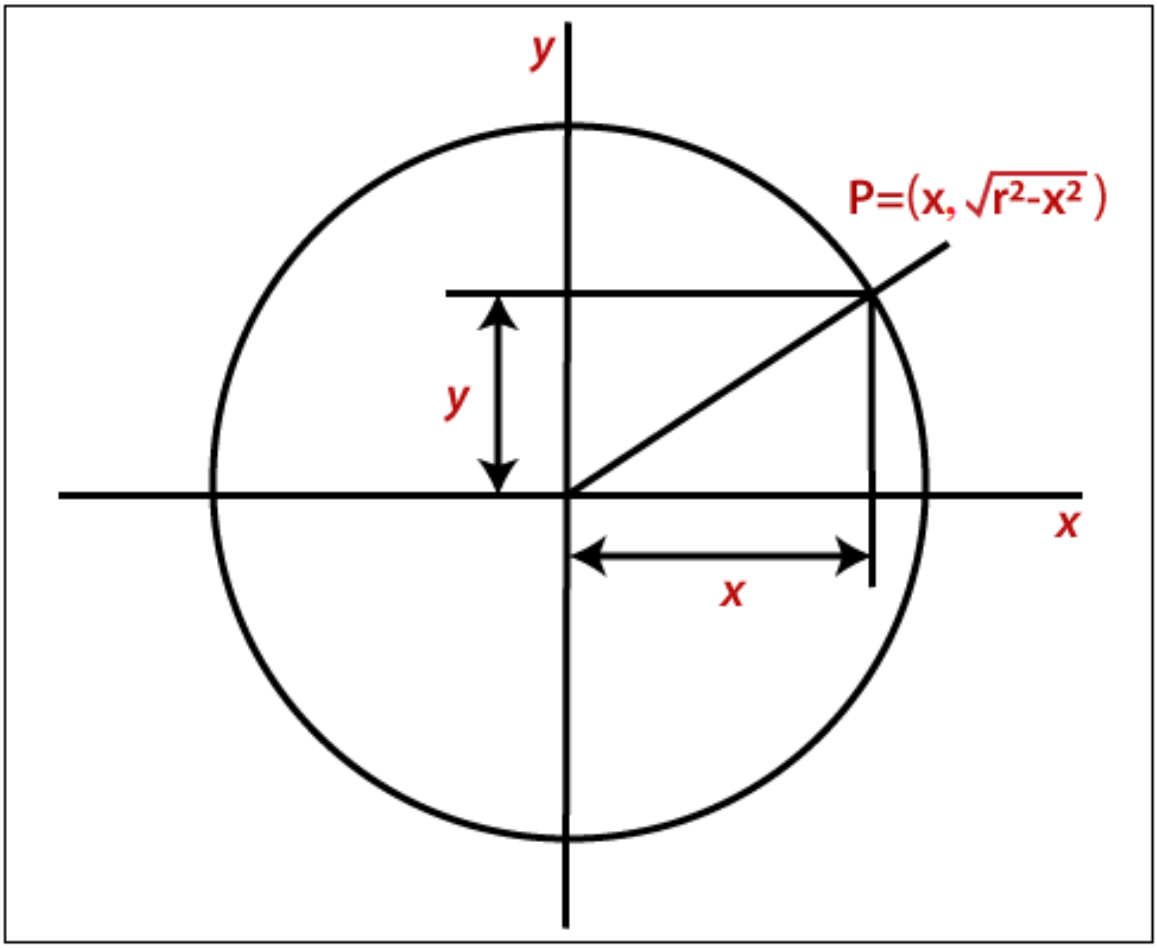
Here, x = The coordinates of x

y = The coordinates of y

r = The radius of the circle

In this method, we will find the x coordinate (90° to 45°) by moving x from 0 to r/√2, and we will find each y coordinate by calculating $\sqrt{r^2-x^2}$ for each step.

It is an ineffective method because for each point x coordinate and radius r must be squaredand subtracted from each other.



A circle with trigonometric/polar coordinate–

$x = r \cos \theta$

$y = r \sin \theta$

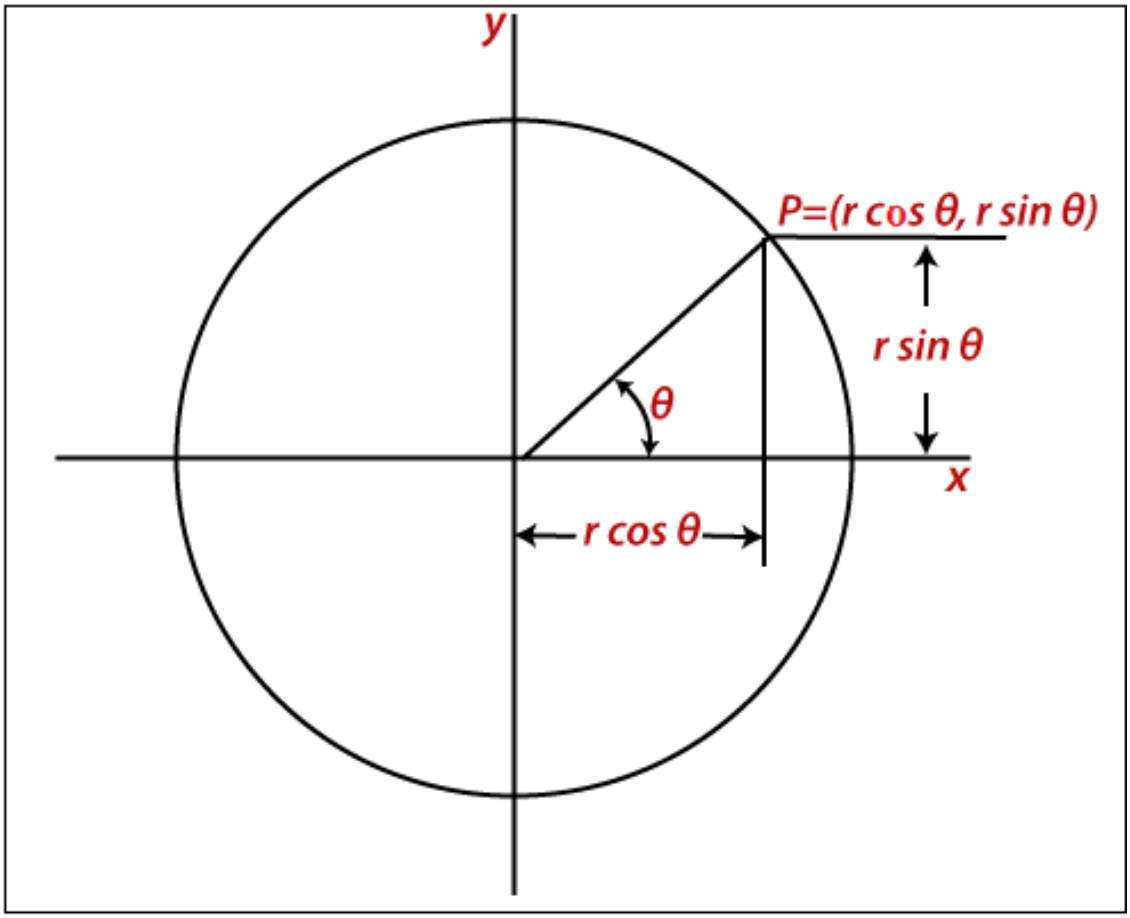
Here, x = The coordinate of x

y = The coordinate of y

r = The radius of the circle

θ = Current angle

In this method, the value of θ moves from 0 to π/4. We will calculate each value of x and y.



Algorithms used in Circle Drawing

There are the following two algorithms used to draw a circle.

- Bresenham's Circle drawing Algorithm
- Mid-point Circle Drawing Algorithm

Scan Converting Ellipses

An ellipse can be defined as a closed plane curve, which is a collection of points. It is the cause of the intersection of a plane over a cone. The ellipse is a symmetric shape figure. It is similar to a circle, but it contains four-way symmetry. We can also define an ellipse as a closed curve created by the points moving in such an order that the total of its distance from two points is constant. These two points are also called "foci."

Properties of Ellipse

- The circle and ellipse are examples of the conic section.
- We can create an ellipse by performing stretching on a circle in x and y-direction.
- Every ellipse has two focal points, called “foci.” In the standard form, we can write the ellipse equation as follow-

$$(x – h)^2/ a^2 + (y – h)^2/ b^2 = 1$$

(h, k) are the center coordinates of the circle.

There should be two axes of an ellipse.

- **Major axis**
- **Minor axis**

Major axis: The major axis is the longest width. For a horizontal ellipse, the major axis and x-axis are parallel to each other. 2a is the length of the major axis. The endpoints of the major axis are vertices with (h ±a, k).

Minor axis: The minor axis is the shortest width of it. For a horizontal ellipse, the minor axis is parallel to the y-axis. 2b is the length of the minor axis. The endpoints of the minor axis are vertices with (h, k ±a).

There are following two methods to define an ellipse-

- **An ellipse with the polynomial method**
- **An ellipse with the trigonometric method**

An Ellipse with Polynomial method

When we use polynomial method to define an ellipse, we will use the following equation-

$$(x – h)^2/ a^2 + (y – k)^2/ b^2 = 1$$

Here, (h, k) = The center points of ellipse

a = Length of the major axis

b = Length of the minor axis

We will use (p, q) for major and minor axes. Now the equation will be-

$$(x - h)^2 / p^2 + (y - k)^2 / q^2 = 1$$

In the polynomial method, the value of the x-axis is increased from the center point (h) to major axis length (p). We will find the value of y-axis by following formula/equation-

$$y = q \sqrt{1- (x - h)^2 + k / p^2}$$

An Ellipse with Trigonometric method

When we use the trigonometric method to define an ellipse, we will use the following equation-

$$x = a \cos (\theta) + h$$

$$y = b \sin (\theta) + h$$

Here, (x, y) = The current coordinates

(h, k) = The center points of ellipse

p = Length of the major axis

q = Length of the minor axis

θ = Current angle

Now, the equation will be-

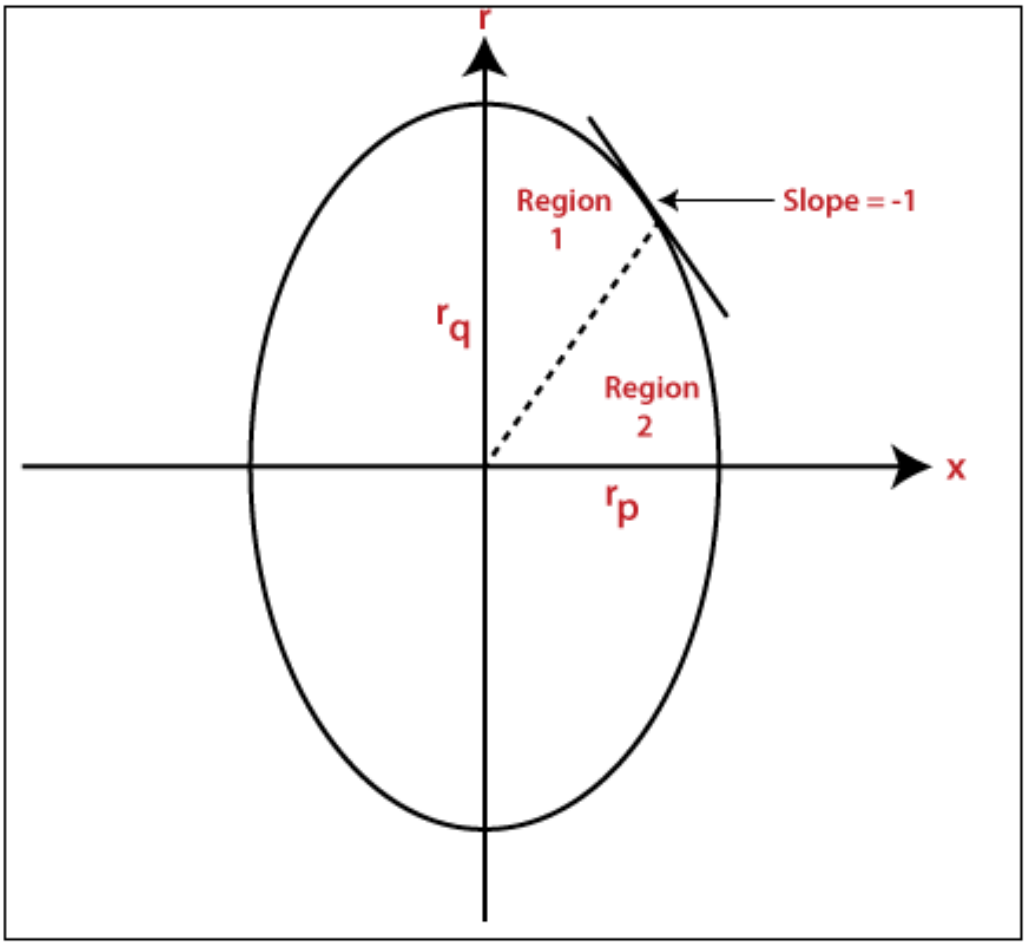
$$x = p \cos (\theta) + h$$

$$y = q \sin (\theta) + h$$

In the trigonometric method, the value of the angle (θ) tends from 0 to π/2 radians. We will find the value of all points by symmetry.

Midpoint Ellipse Drawing Algorithm

We can understand ellipse as an elongated circle. In the midpoint ellipse drawing algorithm, we will determine the plotting point (p, q). Let us suppose that the center point of the ellipse is (0, 0).



The equation of ellipse = $p^2 / r_p^2 + q^2 / r_q^2 = 1$

We will simplify the equation as follows-

$$p^2 r_q^2 + q^2 r_p^2 = r_p^2 r_q^2$$

We can represent the ellipse by following equation-

$$f(p, q) = r_q^2 p^2 + r_p^2 q^2 - r_p^2 r_q^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Now, there are three following conditions-

If

$f(p, q) < 0$

then

(p, q) is inside the ellipse boundary.

If

$f(p, q) = 0$

then

(p, q) is on the ellipse boundary.

If

$f(p, q) > 0$

then

(p, q) is outside the ellipse boundary.

Now,we partially differentiate the equation (1), and we get-

$d_p / d_q = -2r_q^2 p / 2r_p^2 q$

$2r_q^2 p >= 2r_p^2 q$ {when we switch from region 1 to region 2}

Now, we will calculate the decision parameter (d_k1) for region 1–

$d_k1 = (p_k+1, q_k - 1/2)$

Now, we put the value in equation (1)

$f(p, q) = r_q^2 (p_{k+1})^2 + r_p^2 (q_k - 1/2)^2 - r_p^2 r_q^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$

Now, we calculate the next decision parameter ($d_{k1}+1$)

$d_k1+1 = (p_k+1+1, q_k+1 - 1/2)$

Now, we put the value of d_k1+1 in equation (2)

$d_k1+1 = r_q^2 ((p_k+1) +1)^2 + r_p^2 (q_k+1 - 1/2)^2 - r_p^2 r_q^2$

We simplify the equation (2) by $\{(A+B)^2\}$ and $\{(A-B)^2\}$

$d_k1+1 = r_q^2 [(p_k+1)^2 +12+2 (p_k+1)]^2 + r_p^2 [(q_k+1)^2 +1/4 –2(1/2) q_k+1] – r_p^2 r_q^2$

Now, we find the difference between d_k1+1 and d_k1

$d_k1+1 – d_k1= r_q^2 [(p_k+1)^2+12+2(p_k+1)]^2+r_p^2 [(q_k+1)^2+1/4–q_k+1]–r_p^2r_q^2 – r_q^2((p_k+1) +1)^2 + r_p^2 (q_k+1 – 1/2)^2 – r_p^2 r_q^2$

$d_{k1}+1 = d_{k1}+r_q^2 + 2 r_q^2(p_k+1) +r_p^2 [(q_k-1/2)- (q_k-1/2)^2]$ …… (3)

If

Decision parameter (d_k1) < 0

Then

$d_{k1}+1 = d_{k1} + r_q^2 + 2 r_q^2 (p_k+1)$

If

Decision parameter (d_k1) > 0

Then

$d_{k1}+1 = d_{k1} + r_q^2 + 2 r_q^2 (p_k+1) – 2 r_p^2 q_k+1$

Now, we calculate the initial decision parameter $d_{k1}(0)$

$d_{k1}(0) = r_q^2 + ¼ r_p^2 – r_p^2 r_q$ {For region 1}

Thus, we calculate the decision parameter (d_{k2}) for region 2–

$d_k2 = (p_k+1/2, q_k – 1)$

Now, we put the value of d_k2 in equation (1)

$d_k2 = r_q^2 (p_k +1/2)^2 + r_p^2 (q_k -1)^2 – r_p^2 r_q^2$

Now, we calculate the next decision parameter (d_k2+1)

$(d_k2+1) = (p_k+1 +1/2, q_k+1 – 1)$

Now, we put the value of dk2+2 in previous equation

$d_k2+1 = r_q^2 ((p_k+1) +1/2)^2 + r_p^2 (q_k-1) – 1)^2 – r_p^2 r_q^2$

We simplify the equation and calculate the difference between d_k2+1 and d_k2. We get-

$d_k2+1 = d_k2-2r_p^2 (q_k -1) +r_p^2 +r_q^2\ 2\ [(p_k+1+1/2)^2– (p_k+1/2)^2]$ …… (4)

Now, we check two conditions-

If

Decision parameter (d_k2) < 0

Then

$d_k2+1 = d_k2 -2\ r_p^2 q_k+1 + r_p^2$

If

Decision parameter (d_k2) > 0

Then

$d_k2+1 = d_k2 +2r_q^2\ p_k+1 –2\ r_p^2\ q_k+1 +r_p^2$

Now, we calculate the initial decision parameter $d_k2(0)$

$d_k1(0) = r_q^2(p_k +1/2)^2 + r_p^2(q_k –1) – r_p^2 r_q$ {For region 2}

Algorithm of Midpoint Ellipse Drawing

Step 1: First, we read r_p and r_q.

Step 2: Now we initialize starting coordinates as follow-

$p = 0$ and $q = r_q$

Step 3: Wecalculate the initial value and decision parameter for region 1 as follow-

$d_k1 = r^2q – r^2p\ r^2q + ¼\ r^2p$

Step 4: Now, we initialize d_p and d_q as follows-

$d_p = 2r^2qp$

$d_q = 2r^2pq$

Step: 5 Now, we apply Do-while loop

Do

{plot (p, q)

If ($d_k1 < 0$)

(p= p+1 and q = q)

$d_p = d_p + 2r^2q$

$d_k1 = d_k1 – d_p + r^2q$

$[d_k1 = d_k1 + 2r^2qp + 2r^2q + r^2q]$
}
Else
(p= p+1 and q = q -1)

$d_p = d_p + 2r^2q$

$d_q = d_q – 2r^2p$

$d_k1 = d_k1 + d_p – d_p + r^2q$

$[d_k1 = d_k1 + 2r^2qp + 2r^2q – (2r^2pq – 2r^2p) + r^2q]$

} while ($d_p < d_q$)

Step 6: Now,wecalculate the initial value and decision parameter for region 2 as follow-

$d_k2 = r^2q\ (p + 1/2)^2 + r^2p\ (q-1)^2 – r^2p\ r^2q$

Step 7: Now, we again apply Do-while loop

Do

{plot (p, q)

If ($d_k2 < 0$)

(p= p and q = q – 1)

$d_q = d_p – 2r^2p$

$d_k2 = d_k2 – d_q + r^2p$

[$d_k2 = d_k2 – (2r^2pq – 2r^2p) + r^2p$]
}
Else
(p= p+1 and q = q -1)

$d_q = d_q – 2r^2p$

$d_p = d_p + 2r^2q$

$d_k2 = d_k2 + d_p – d_q + r^2p$

[$d_k2 = d_k2 + 2r^2qp + 2r^2q – (2r^2pq – 2r^2p) + r^2q$]

} while (q < 0)

Step 8: Thus, we calculate all the points of other quadrants.

Step 9: Stop.

UNIT-II

Hardcopy Technologies

Hardcopy is a printed copy of information from a computer. Sometimes it refer to as a printout, so it called hardcopy because it exists as a physical object. Hardcopy is tangible output that usually printed. The principal examples are printouts, whether text or graphics, form printers and also films including microfilms and microfirche is also considered as hardcopy output.

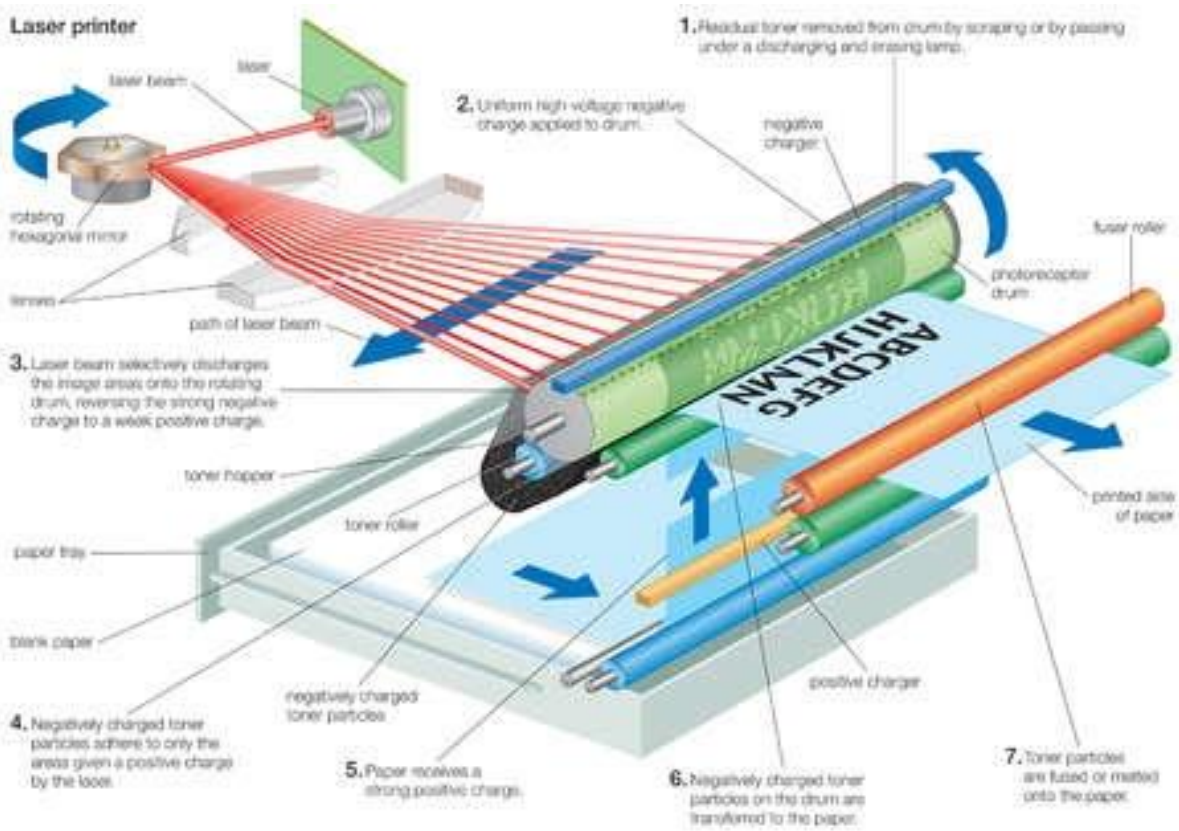
Printers

Printers are the most user use to produce an hardcopy output which is printed paper. Printers can print text and graphics on paper or other hardcopy materials. Printers resolution is measured by dots per inch (dpi): 1,200 times 1,200 is the most common for microcomputers. The printers are classified as two categories either impact printers or non -impact printers. The Impact printers is dot-matrix printer that can print by striking the paper directly. Non-impact printers are such as laser printers and inkjet printers that do not have direct contact with the hardcopy medium.

Types of Impact Printers

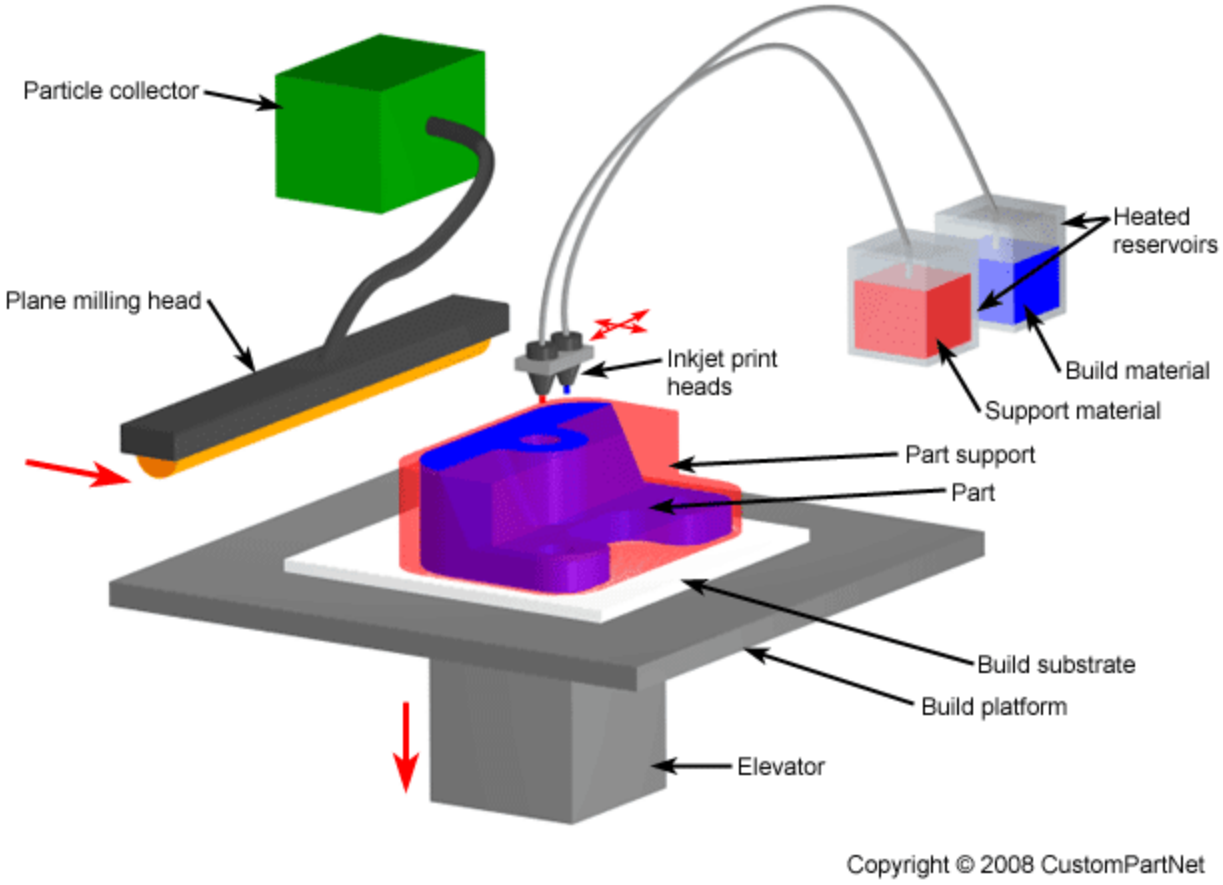
- **Laser Printer**

Laser printer is a dot-matrix printer which creates images with dots. however, as in a photocopying machine, these images are produced on a drum, treated wirt an electrically charged ink like toner which powder compound, and then transferred fromdrum to paper. It runs with software called a page description language (PDL), which tells the printer how to lay out the printed page and support various fonts. Laser printer normally have their own CPU, ROM, and memory to stored information data for a while before printing on paper to prevent the lost of data while printing session. Thegraphics-heavy document required much more memory compared to texts ordocuments.



- **Inkjet Printer**

Inkjet printers work by spray ink onto paper small that electrically charged droplets of inkfrom four or more nozzles through holes in a matrix at high speed. Alike laser and dot matrix printers, inkjet printers form images with little dots which produce better output. Inkjet printers commonly have a dpi of 4,800 times 1,200 makes it can spray ink onto the page a line at a time. They have high quality in both high-quality black and white text and high-quality colour graphics.



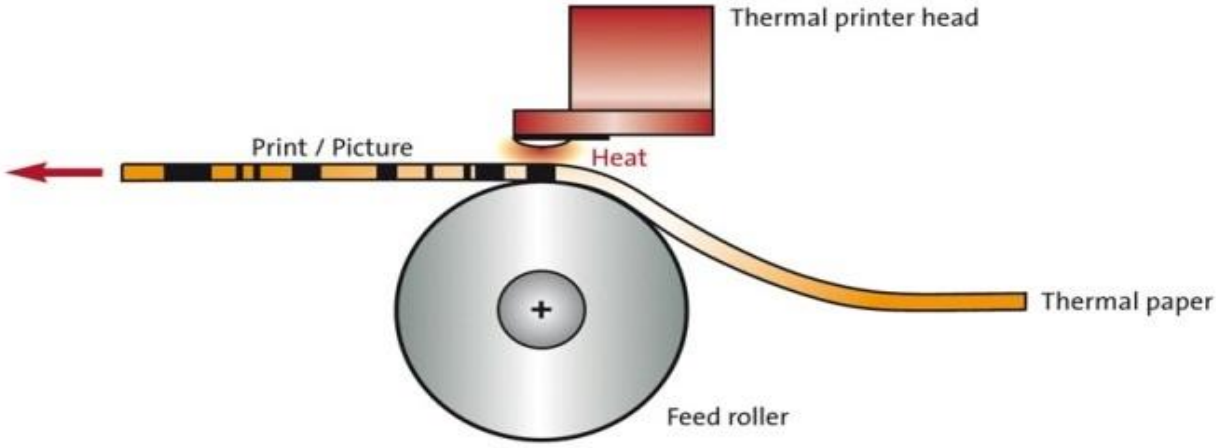
Inkjet printer process

Example of inkjet printer

Types of Non-Impact Printers

- **Thermal Printers**

Thermal printers is a low to medium resolution printers that use a type of coated paperthat darkens when heat is applied to it. It typically used in business for bar-code label applications and for printing cash register receipts.

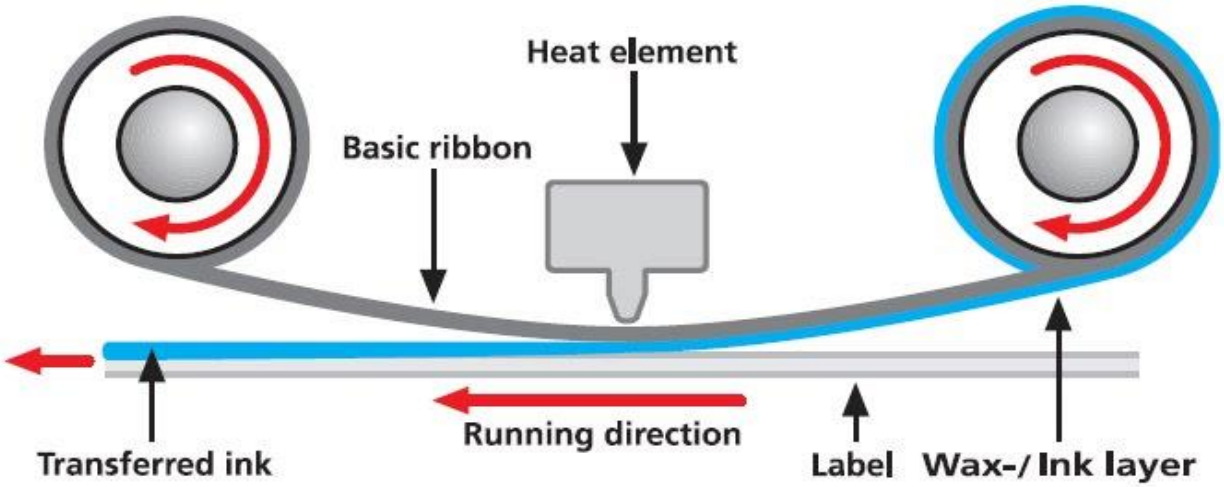


Thermal printer process

Example of thermal printer

- **Thermal Wax-Transfer Printers**

The thermal wax-transfer printer are use to print a wax-based ink onto paper. After it becomes cool, the wax adheres permanently to the paper. because of their water fastness, these labels find uses in industrial label printing.



Thermal wax-transfer process

Example of thermal wax-transfer printer

- **Photo Printers**

Photo printer is a specialized machines for printing continuous-tone photo prints withspecial paper and colour dyes. Usually photo printer are much expensive compared toothers.

Example of photo printer

Multifunction Printers

The multifunction printers are actually is a combination of all printers where they can do printing, scanning, copying and also faxing. The advantages by using this kind of printers are cost less and take up less space to stored rather than buying the four separated printers. But, it comes a problems when one component are malfunction, the other will malfunction also. Most office are using this kind of printers to do their jobs easily.

Example of multifunction printer

Plotters

Generally, plotters are used in designed for large-format printing where they are specialized output device designed to produce large high-quality 3D graphics in a variety of colour. It commonly used in fields of architecture, engineering, and map-makers. Plotter can be classified into three which are pen plotters, electrostatic plotters, and large-format plotters. Pen plotters refer of use one or more coloured pens. Electrostatic plotters are lie partially flat on a table and use toner as photocopiers do. Large-format plotter are large-scale inkjet printers used by graphic artists.

Display Technologies

The display device is an output device used to represent the information in the form of images (visual form). Display systems are mostly called a video monitor **or** Video display unit (VDU).

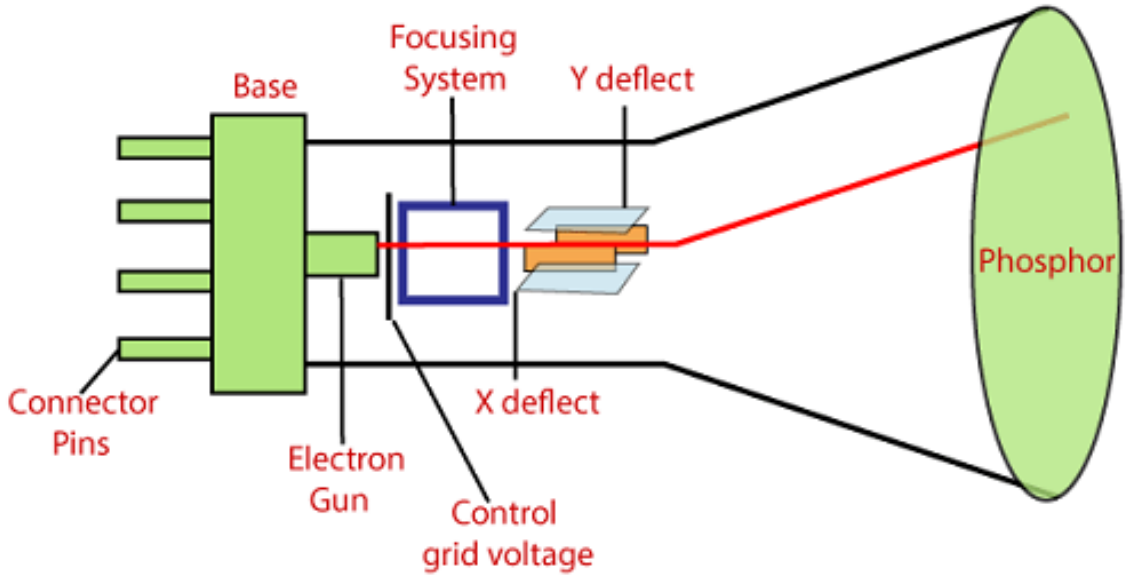
Display devices are designed to model, display, view, or display information. The purpose of display technology is to simplify information sharing.

Today, the demand for high-quality displays is increasing.

There are some display devices given below:

1. Cathode-Ray Tube(CRT)
2. Color CRT Monitor
3. Liquid crystal display(LCD)
4. Light Emitting Diode(LED)
5. Direct View Storage Tubes(DVST)
6. Plasma Display
7. 3D Display

1. **Cathode-ray Tube (CRT):** Here, CRT stands for Cathode ray tube. It is a technology which is used in traditional computer monitor and television.
   Cathode ray tube is a particular type of vacuum tube that displays images when an electron beam collides on the radiant surface.



Component of CRT

- **Electron Gun:** The electron gun is made up of several elements, mainly a heating filament (heater) and a cathode. The electron gun is a source of electrons focused on a narrow beam facing the CRT.

- **Focusing & Accelerating Anodes:** These anodes are used to produce a narrow and sharply focused beam of electrons.

- **Horizontal & Vertical Deflection Plates:** These plates are used to guide the path of the electron the beam. The plates produce an electromagnetic field that bends the electron beam through the area as it travels.

- **Phosphorus-coated Screen:** The phosphorus coated screen is used to produce bright spots when the high-velocity electron beam hits it.

There are two ways to represent an object on the screen:

1. **Raster Scan:** It is a scanning technique in which the electron beam moves along the screen. It moves from top to bottom, covering one line at a time.

   A raster scan is based on pixel intensity control display as a rectangular box on the screen called a raster.

   Picture description is stored in the memory area called as Refresh buffer, or Frame Buffer.

   Frame buffer is also known as Raster or Bitmap**.** Raster scan provides the refresh rate of 60 to 80 frames per second.

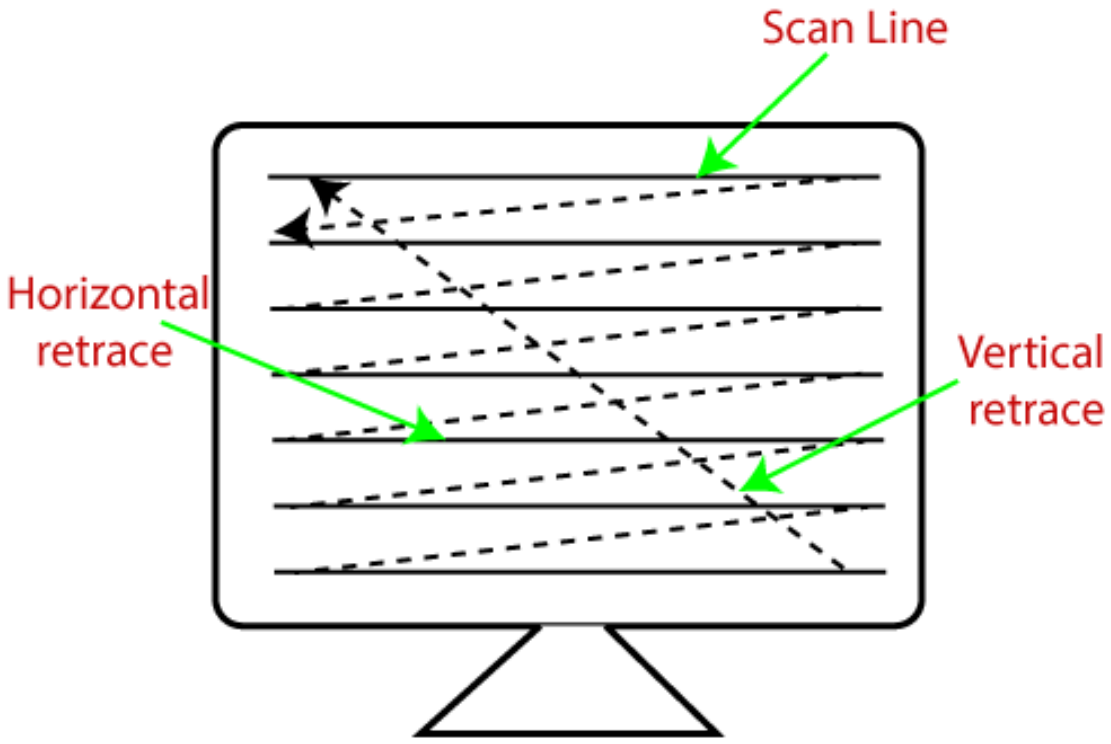
   For Example: Television

   The beam refreshing has two types:

1. Horizontal Retracing
2. Vertical Retracing

   When the beam starts from the top left corner and reaches bottom right, and again return to the top left, it is called the vertical retrace.

   It will call back from top to bottom more horizontally as a horizontal reversal.



Advantages:

1. Real image
2. Many colors to be produced
3. Dark scenes can be pictured

Disadvantages:

1. Less resolution
2. Display picture line by line
3. More costly

2. Random Scan (Vector scan): It is also known as stroke-writing display or calligraphic display. In this, the electron beam points only to the area in which the picture is to be drawn.

It uses an electron beam like a pencil to make a line image on the screen. The image is constructed from a sequence of straight-line segments. On the screen, each line segment is drawn by the beam to pass from one point on the screen to the other, where its x & y coordinates define each point.

After compilation of picture drawing, the system cycle back to the first line and create all the lines of picture 30 to 60 times per second.

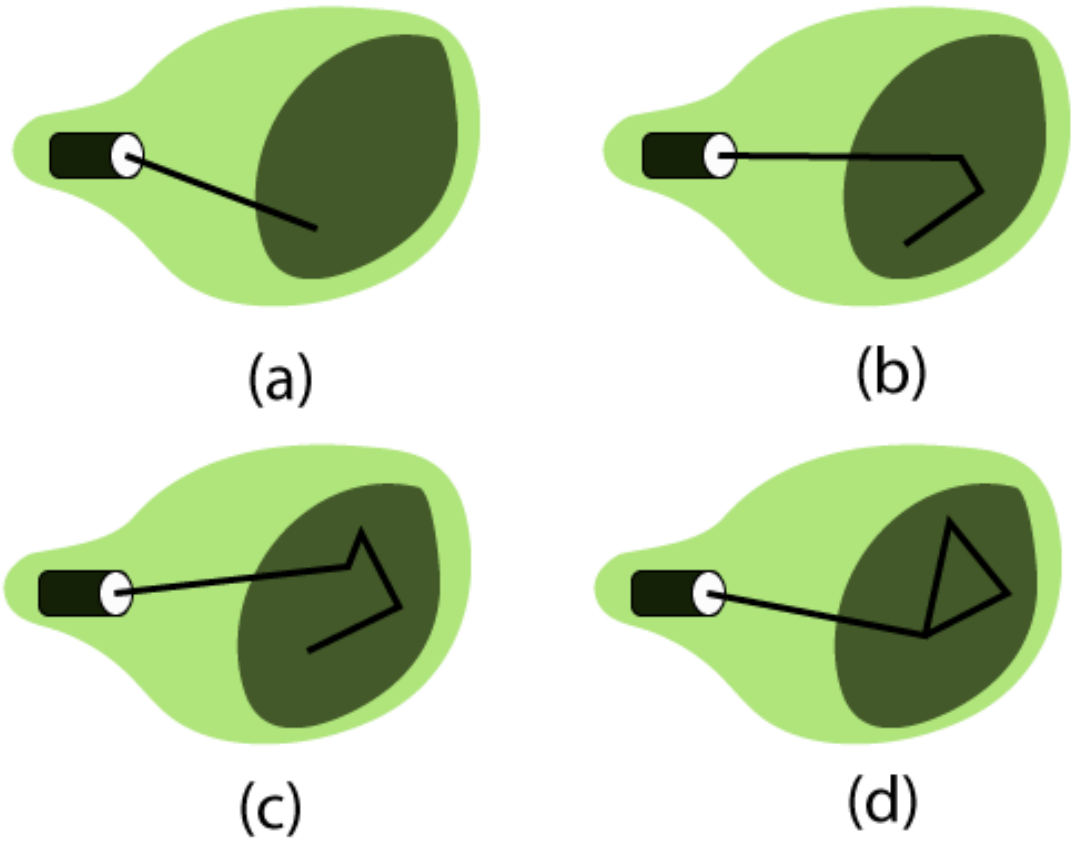


Fig: A Random Scan display draws the lines of an object in a specific order

Advantages:

1. High Resolution
2. Draw smooth line Drawing

Disadvantages:

1. It does only the wireframe.
2. It creates complex scenes due to flicker.

2. **Color CRT Monitor:** It is similar to a CRT monitor.
The basic idea behind the color CRT monitor is to combine three basic colors- Red, Green, and Blue. By using these three colors, we can produce millions of different colors.

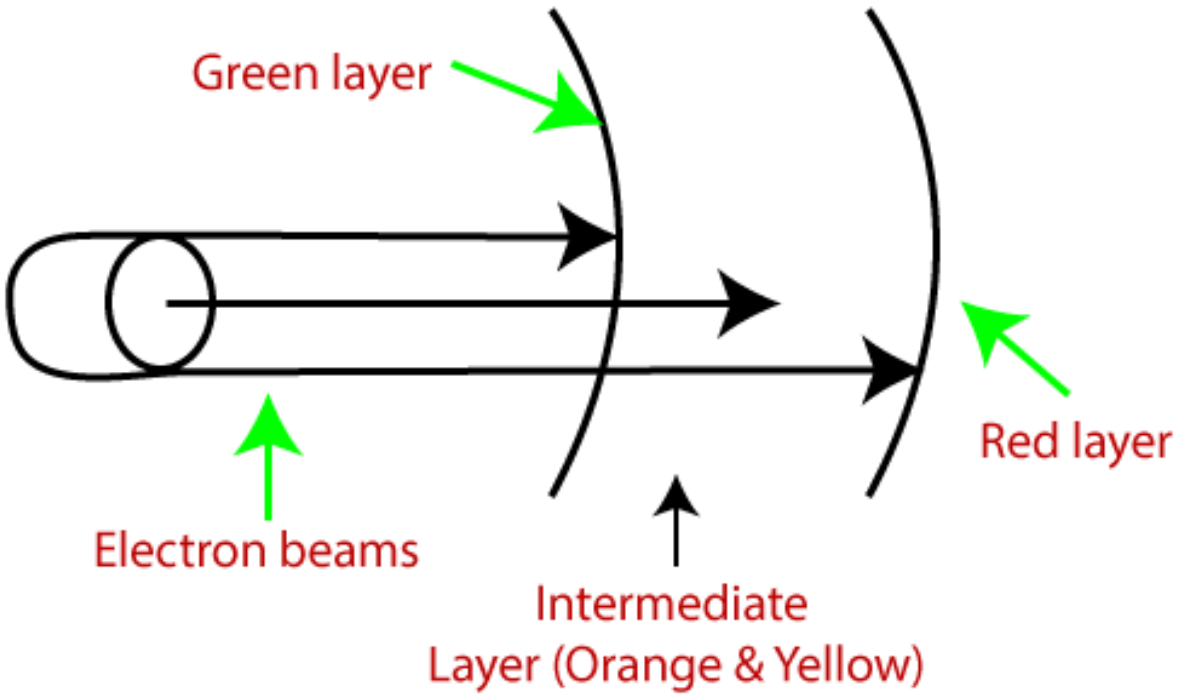
The two basic color display producing techniques are:

1. **Beam–Penetration Method:** It is used with a random scan monitor for displaying pictures. There are two phosphorus layers- Red and Green are coated inside the screen. The color shown depends on how far the electron beam penetrates the phosphorus surface.

   A powerful electron beam penetrates the CRT, it passes through the red layer and excites the green layer within.

   A beam with slow electrons excites only the red layer.

   A beam with the medium speed of electrons, a mixture of red and green light is emitted to display two more colors- orange and yellow.



Advantages:

1. Better Resolution
2. Half cost
3. Inexpensive

Disadvantages:

1. Only four possible colors
2. Time Consuming

2. Shadow–Mask Method: It is used with a raster scan monitor for displaying pictures. It has more range of color than the beam penetration method. It is used in television sets and monitors.

Structure:

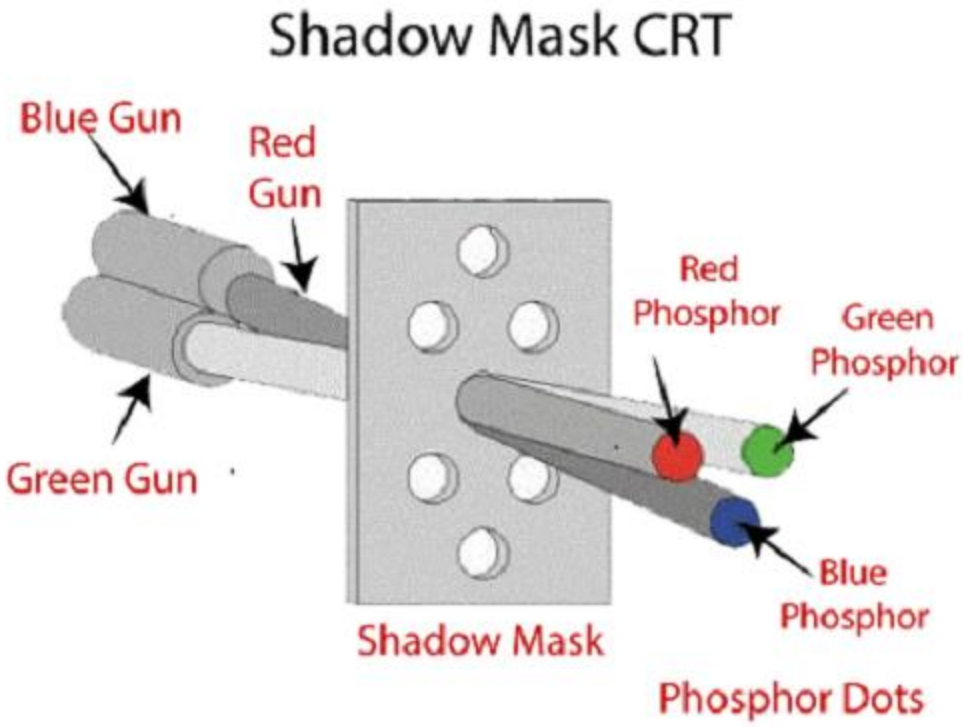
1. It has three phosphorus color dots at each position of the pixel.
   First Dot: Red color
   Second Dot: Green color
   Third Dot: Blue color

1. It has three different guns. Each for one color.
2. It has a metal screen or plate just before the phosphorus screen, named **“**Shadow-Mask.”
3. It also has a shadow grid just behind the phosphorus coated screen with tiny holes in a triangular shape.

Working: A Shadow Mask is a metal plate with tiny holes present inside a color monitor.

A Shadow Mask directs the beam by consuming the electrons so that the beam hits only the desired point and displays a resulting picture.

It has three different guns. These guns direct their beams to shadow mask, which allows them to pass. It is a task of a shadow mask to direct the beam on its particular dot on the screen and produce a picture on the screen.
A Shadow Mask can display a wider range of pictures than beam penetration.



Advantages:

1. Display a wider range picture.
2. Display realistic images.
3. In-line arrangement of RGB color.

Disadvantages:

1. Difficult to cover all three beams on the same hole.
2. Poor Resolution.

3. **Liquid crystal display (LCD):** The LCD depends upon the light modulating properties of liquid crystals.

LCD is used in watches and portable computers. LCD requires an AC power supply instead of DC, so it is difficult to use it in circuits.

It generally works on flat panel display technology. LCD consumes less power than LED. The LCD screen uses the liquid crystal to turn pixels on or off.
Liquid Crystals are a mixture of solid and liquid. When the current flows inside it, its position changes into the desired color.

For Example: TFT (Thin Film Transistor)

Advantages:

1. Produce a bright image
2. Energy efficient
3. Completely flat screen

Disadvantages:

1. Fixed aspect ratio & Resolution
2. Lower Contrast
3. More Expensive

4. Light Emitting Diode (LED): LED is a device which emits when current passes through it. It is a semiconductor device.

The size of the LED is small, so we can easily make any display unit by arranging a large number of LEDs.

LED consumes more power compared to LCD. LED is used on TV, smartphones, motor vehicles, traffic light, etc.

LEDs are powerful in structure, so they are capable of withstanding mechanical pressure. LED also works at high temperatures.

Advantages:

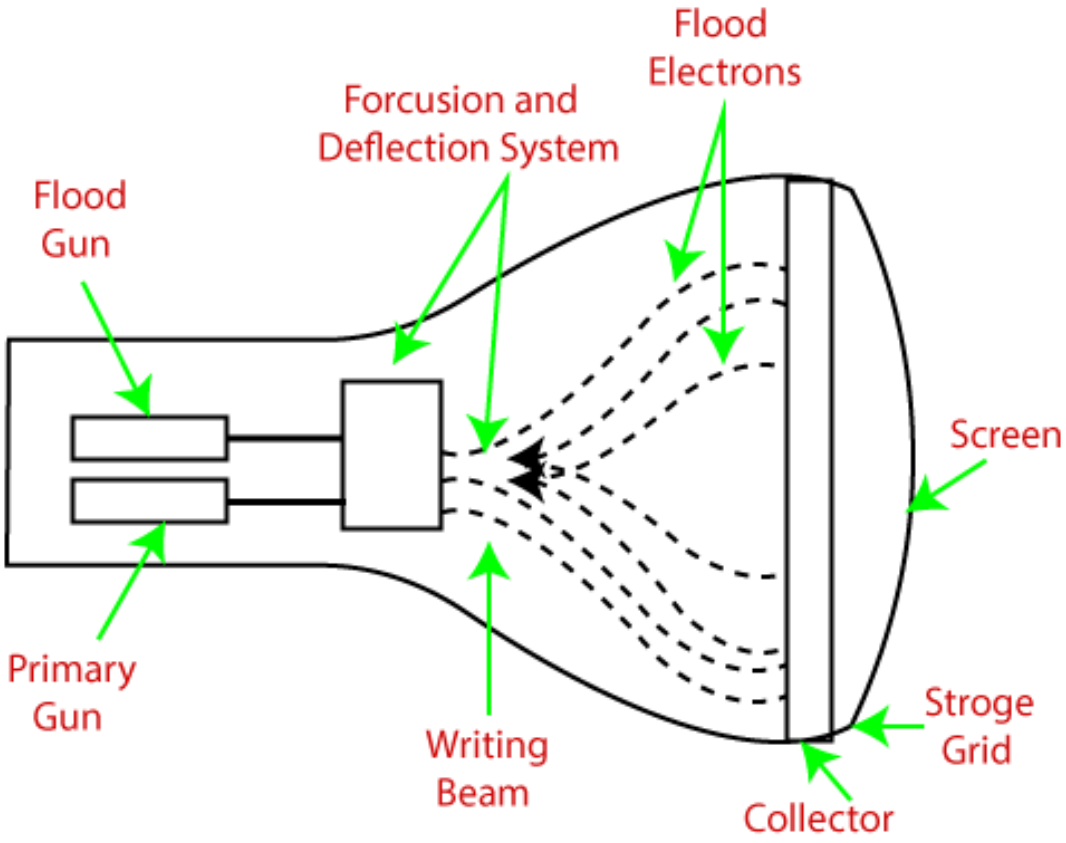
1. The Intensity of light can be controlled.
2. Low operational Voltage.
3. Capable of handling the high temperature.

Disadvantages:

1. More Power Consuming than LCD.

5. Direct View Storage Tube (DVST): It is used to store the picture information as a charge distribution behind the phosphor-coated screen.
There are two guns used in DVST:

1. **Primary Gun:** It is used to store the picture information.
2. **Flood / Secondary Gun:** It is used to display a picture on the screen.



Advantages:

1. Less Time Consuming
2. No Refreshing Required
3. High-Resolution
4. Less Cost

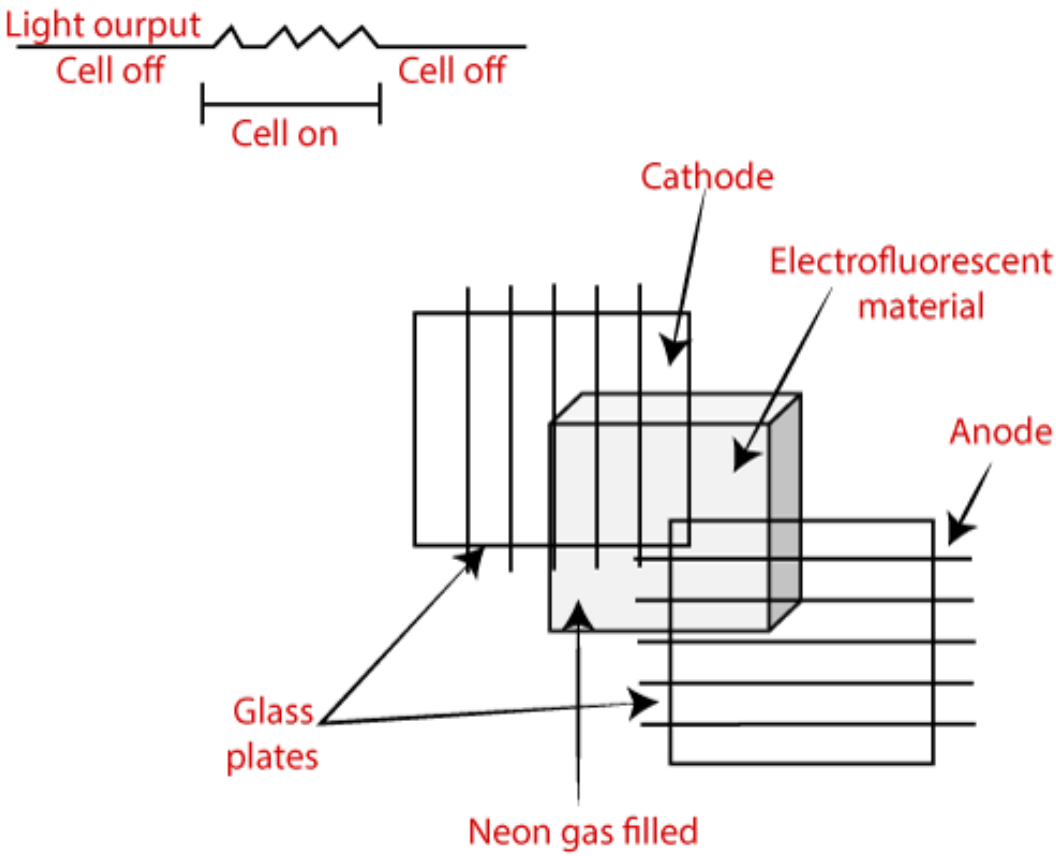
Disadvantages:

- The specific part of the image cannot be erased.
- They do not display color.

6. Plasma Display: It is a type of flat panel display which uses tiny plasma cells. It is also known as the Gas-Discharge display.

Components of Plasma display

1. **Anode:** It is used to deliver a positive voltage. It also has the line wires.
2. **Cathode:** It is used to provide negative voltage to gas cells. It also has fine wires.
3. **Gas Plates:** These plates work as capacitors. When we pass the voltage, the cell lights regularly.
4. **Fluorescent cells:** It contains small pockets of gas liquids when the voltage is passed to this neon gas. It emits light.



Advantages:

1. Wall Mounted
2. Slim
3. Wider angle

Disadvantages:

1. Phosphorus loses luminosity over time.

2. It consumes more electricity than LCD.
3. Large Size

7. 3D Display: It is also called stereoscope display technology. This technology is capable of bringing depth perception to the viewer.

It is used for 3D gaming and 3D TVs.

For Example: Fog Display, Holographic Display, Retina Display Etc.

Advantages:

- Impressive Picture Quality
- Impressive Picture Quality
- Impressive Picture Quality

Disadvantage:

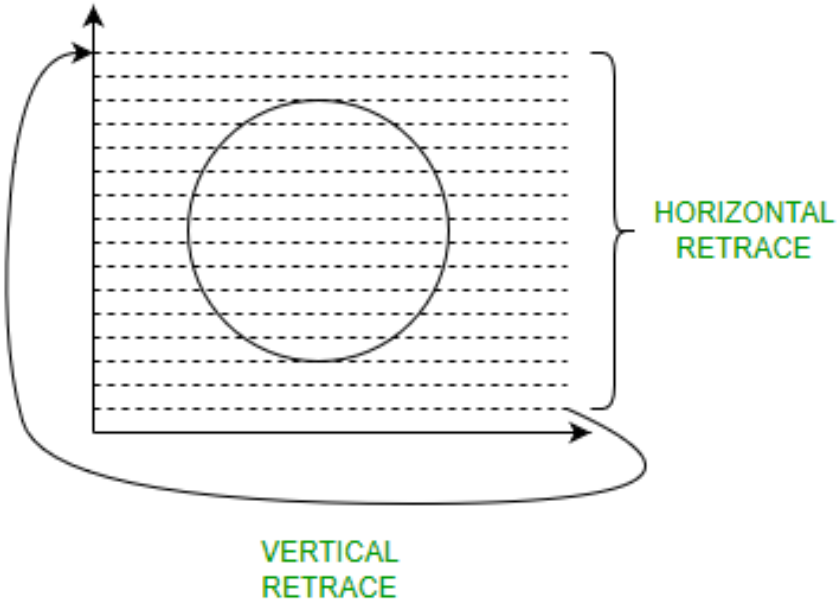
- Expensive
- Binocular Fusion

Example of plotters

Raster-Scan Display System

Raster Scan Displays are most common type of graphics monitor which employs CRT. It is based on television technology. In raster scan system electron beam sweeps across the screen, from top to bottom covering one row at a time.A pattern of illuminated pattern of spots is created by turning beam intensity on and off as it moves across each row. A memory area called refresh buffer or frame buffer stores picture definition. This memory area holds intensity values for all screen points. Stored intensity values are restored from frame buffer and painted on screen taking one row at a time.Each screen point is referred to as pixels.

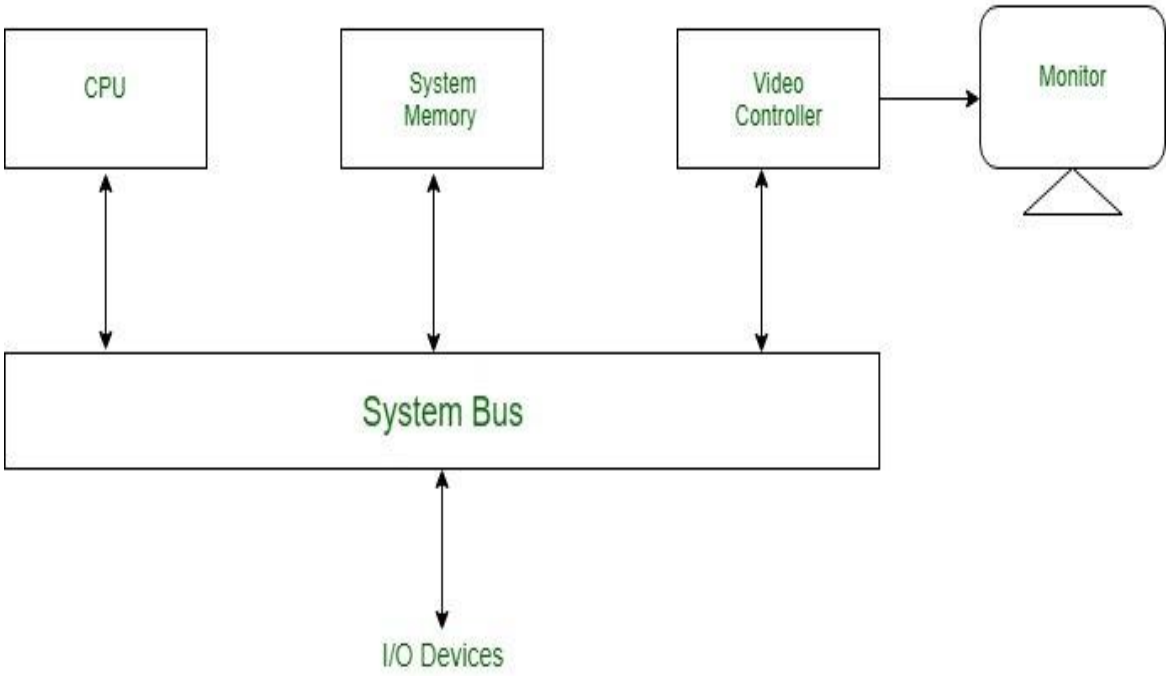
In raster scan systems refreshing is done at done at a rate of 60-80 frames per second. Refresh rates are also sometimes described in units of cycles per second / Hertz (Hz). At the end of each scan line, electron beam begins to display next scan line after returning to left side of screen. The return to the left of screen after refresh of each scan line is known as *horizontal retrace* of electron beam. At the end of each frame electron beam returns to top left corner and begins the next frame.



Raster-Scan Display Processor:

An important function of display process is to digitize a picture definition given in an application program into a set of pixel-intensity values for storage in refresh buffer. This process is referred to as scan conversion. The purpose of display processors is to relieve the CPU from graphics jobs.

Display processors can perform various other tasks like: creating different line styles, displaying color areas, etc. Typically display processors are utilized to interface input devices, such as mouse, joysticks.



ADVANTAGES:

- Real life images with different shades can be displayed.
- Color range available is bigger than random scan display.

DISADVANTAGES:

- Resolution is lower than random scan display.
- More memory is required.
- Data about the intensities of all pixel has to be stored.

Video Controller

A video controller, often referred to as a video or graphics card, is a key hardware component that allows computers to generate graphic information to any video display devices, such as a monitor or projector. They are also known as graphics or video adapters. Some modern computers do not include video cards, but rather have graphics processing units directly integrated into the computer's motherboard.

Older Video Controllers

A video controller, once more commonly referred to as a video display controller, were used in older models of home-computers during the 1980s; they were also used in some early video game system consoles. Their main function as an integrated circuit in a video signal generator was to produce television video signals in computers or game systems. Although they could generate graphics, older video controller models did not have specialized hardware accelerators that created 2D and 3D images.

Evolution of the Video Controller

Modern video controllers are installed with hardware accelerators that create both 2D and 3D images. They also offer various functions beyond accelerated image rendering, such as TV output and the ability to hook up to several monitors. Although many computers' motherboards are already integrated with graphics processing units, you can disable the integrated graphics chip via the computer's BIOS to install a higher-performance video controller via the accelerated graphics port. For a modern video controller to function properly in a computer, a computer needs to have four essential units: a functioning motherboard, a processor that generates the power that a video controller needs to perform its tasks, enough memory to distribute the images created by the GPU and a screen or monitor to properly display these images.

GPU

As the brain of a computer's motherboard is the CPU, video controllers have their own unique "centers," referred to as the graphics processing unit, although the GPU is also referred to as the visual processing unit. The GPU's specialized electronic circuit is designed specifically to translate data into graphic images and performs complex mathematical calculations in order to do so. GPUs are also embedded into mobile phones and game consoles.

Advanced Video Cards

A modern video controller, more frequently referred to as a video card, are installed into expansion slots onto the motherboard of a computer. The parts of a modern video card include power supply connectors, a cooling fan, a GPU, and typically also have a PCIe interface, Graphics Double Data Rate version 5 memory, a display port, a digital video interface and an HDMI interface. While some video cards have only one port for connection, other advanced cards have multiple ports that connect to additional televisions and monitors.

Advanced 3D graphics cards, which are more expensive than the average consumer graphics card, allow consumers to preview modeling viewpoints more fluidly. For example, both AMD Radeon and Nvidia release popular graphics cards used by gamers. At the time of publication, the specs for a high-performance video card made by AMD is the Radeon video card, which has 4GB of memory, 1250 MHz memory clock speed and 320GB per-second memory bandwidth.

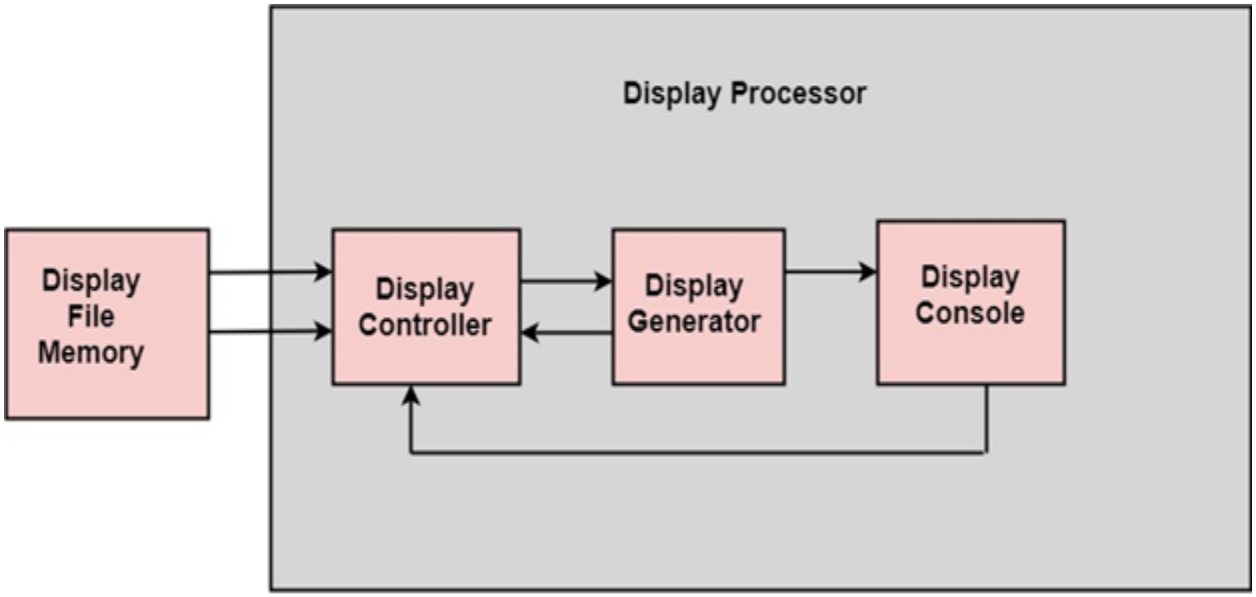
For graphic artists, many computers come with GPU-accelerated apps, such as Microsoft's DirectX or Nvida's close integration with the Autodesk suite. Rather than utilizing a video card slot, GPU-accelerated programs are integrated into the CPU.

Random-Scan Display processor

It is interpreter or piece of hardware that converts display processor code into pictures. It is one of the four main parts of the display processor

Parts of Display Processor

1. Display File Memory
2. Display Processor
3. Display Generator
4. Display Console



Block diagram of Display System

Display File Memory: It is used for generation of the picture. It is used for identification of graphic entities.

Display Controller:

1. It handles interrupt
2. It maintains timings
3. It is used for interpretation of instruction.

Display Generator:

1. It is used for the generation of character.
2. It is used for the generation of curves.

Display Console: It contains CRT, Light Pen, and Keyboard and deflection system.

The raster scan system is a combination of some processing units. It consists of the control processing unit (CPU) and a particular processor called a display controller. Display Controller controls the operation of the display device. It is also called a video controller.

Working: The video controller in the output circuitry generates the horizontal and vertical drive signals so that the monitor can sweep. Its beam across the screen during raster scans.

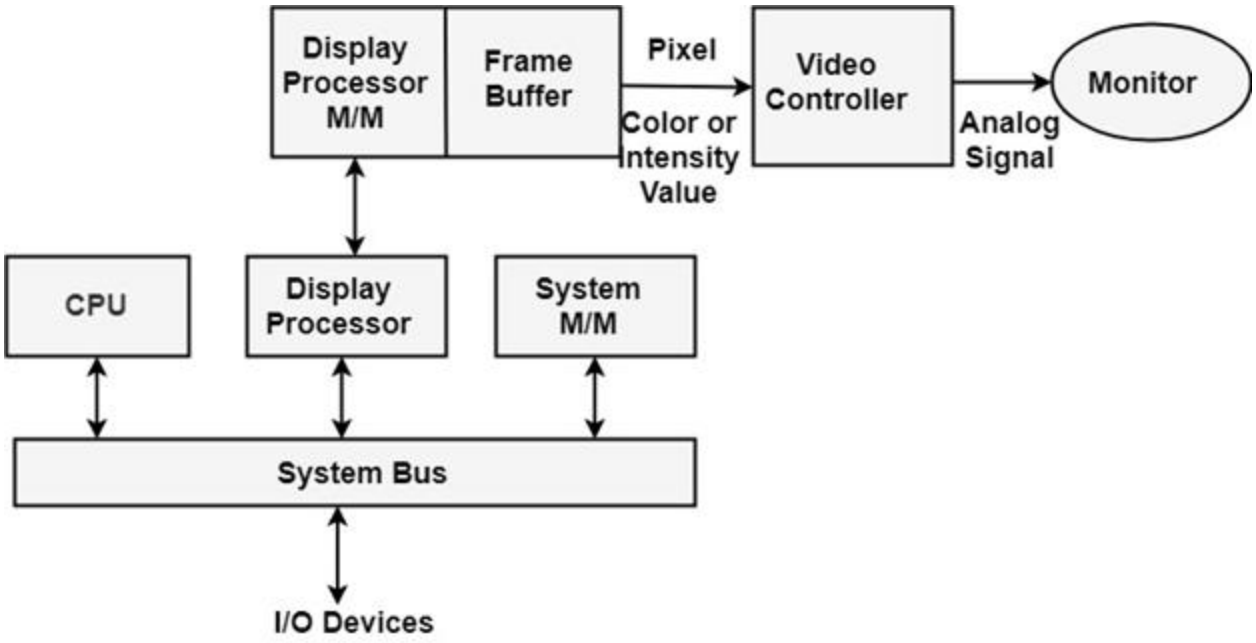


Fig: Architecture of a Raster Display System with a Display Processor

As fig showing that 2 registers (X register and Y register) are used to store the coordinate of the screen pixels. Assume that y values of the adjacent scan lines increased by 1 in an upward direction starting from 0 at the bottom of the screen to y_{max} at the top and along each scan line the screen pixel positions or x values are incremented by 1 from 0 at the leftmost position to x_{max} at the rightmost position.

The origin is at the lowest left corner of the screen as in a standard Cartesian coordinate system.

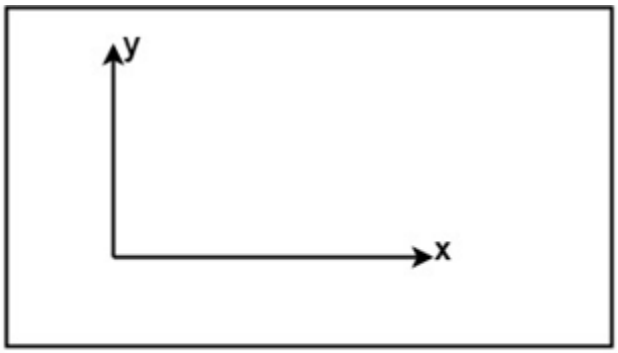


Fig: The origin of the coordinate system for identifying screen positions is usually specified in the lower-left corner.

At the start of a **Refresh Cycle**:

X register is set to 0 and y register is set to y_{max}. This (x, y') address is translated into a memory address of frame buffer where the color value for this pixel position is stored.

The controller receives this color value (a binary no) from the frame buffer, breaks it up into three parts and sends each element to a separate Digital-to-Analog Converter (DAC).

These voltages, in turn, controls the intensity of 3 e-beam that are focused at the (x, y) screen position by the horizontal and vertical drive signals.

This process is repeated for each pixel along the top scan line, each time incrementing the X register by Y.

As pixels on the first scan line are generated, the X register is incremented throughx_{max}.

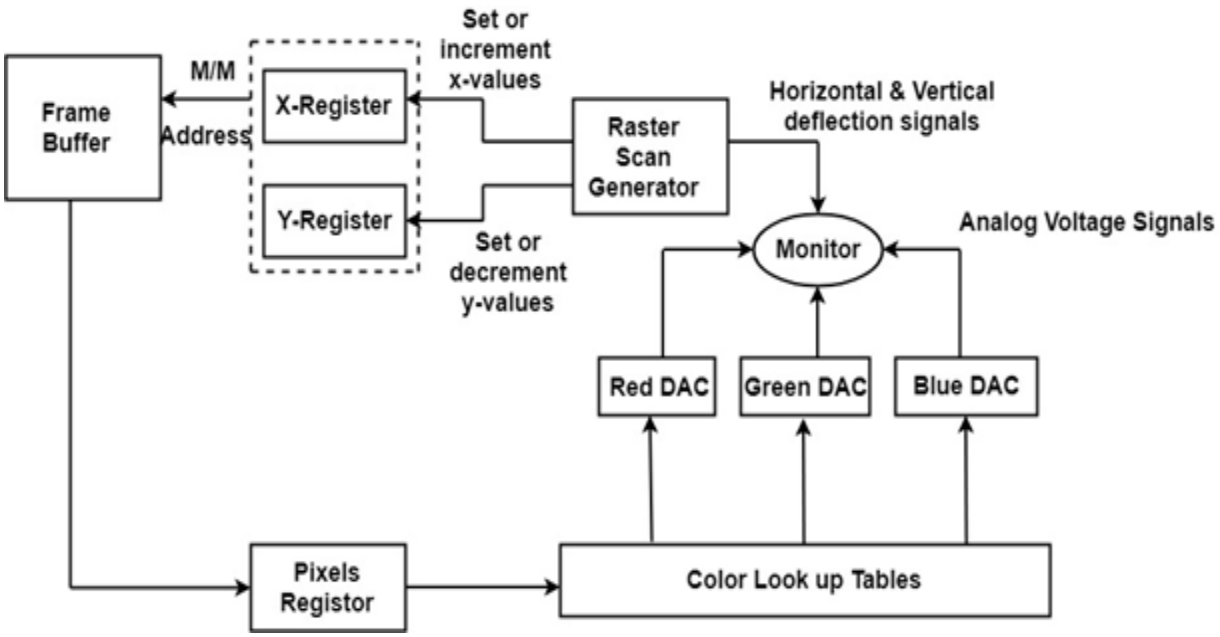
Then x register is reset to 0, and y register is decremented by 1 to access the next scan line.

Pixel along each scan line is then processed, and the procedure is repeated for each successive scan line units pixels on the last scan line (y=0) are generated.

For a display system employing a color look-up table frame buffer value is not directly used to control the CRT beam intensity.

It is used as an index to find the three pixel-color value from the look-up table. This lookup operation is done for each pixel on every display cycle.

As the time available to display or refresh a single pixel in the screen is too less, accessing the frame buffer every time for reading each pixel intensity value would consume more time what is allowed:



Multiple adjacent pixel values are fetched to the frame buffer in single access and stored in the register.

After every allowable time gap, the one-pixel value is shifted out from the register to control the warm intensity for that pixel.

The procedure is repeated with the next block of pixels,and so on, thus the whole group of pixels will be processed.

Display Devices:

The most commonly used display device is a video monitor. The operation of most video monitors based on CRT (Cathode Ray Tube). The following display devices are used:

1. Refresh Cathode Ray Tube
2. Random Scan and Raster Scan
3. Color CRT Monitors
4. Direct View Storage Tubes
5. Flat Panel Display
6. Lookup Table

Input Devices for Operator Interaction

Following are some of the important input devices which are used in a computer −

- Keyboard
- Mouse
- Joy Stick
- Light pen
- Track Ball
- Scanner
- Graphic Tablet
- Microphone
- Magnetic Ink Card Reader(MICR)
- Optical Character Reader(OCR)
- Bar Code Reader
- Optical Mark Reader(OMR)

Keyboard

Keyboard is the most common and very popular input device which helps to input data to the computer. The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing additional functions.

Keyboards are of two sizes 84 keys or 101/102 keys, but now keyboards with 104 keys or 108 keys are also available for Windows and Internet.

The keys on the keyboard are as follows −

S.No	Keys & Description
1	**Typing Keys** These keys include the letter keys (A-Z) and digit keys (09) which generally give the same layout as that of typewriters.
2	**Numeric Keypad** It is used to enter the numeric data or cursor movement. Generally, it consists of a set of 17 keys that are laid out in the same configuration used by most adding machines and calculators.
3	**Function Keys** The twelve function keys are present on the keyboard which are arranged in a row at the top of the keyboard. Each function key has a unique meaning and is used for some specific purpose.
4	**Control keys** These keys provide cursor and screen control. It includes four directional arrow keys. Control keys also include Home, End, Insert, Delete, Page Up, Page Down, Control(Ctrl), Alternate(Alt), Escape(Esc).
5	**Special Purpose Keys** Keyboard also contains some special purpose keys such as Enter, Shift, Caps Lock, Num Lock, Space bar, Tab, and Print Screen.

Mouse

Mouse is the most popular pointing device. It is a very famous cursor-control device having a small palm size box with a round ball at its base, which senses the movement

of the mouse and sends corresponding signals to the CPU when the mouse buttons are pressed.

Generally, it has two buttons called the left and the right button and a wheel is present between the buttons. A mouse can be used to control the position of the cursor on the screen, but it cannot be used to enter text into the computer.

Advantages

- Easy to use
- Not very expensive
- Moves the cursor faster than the arrow keys of the keyboard.

Joystick

Joystick is also a pointing device, which is used to move the cursor position on a monitor screen. It is a stick having a spherical ball at its both lower and upper ends. The lower spherical ball moves in a socket. The joystick can be moved in all four directions.

The function of the joystick is similar to that of a mouse. It is mainly used in Computer Aided Designing (CAD) and playing computer games.

Light Pen

Light pen is a pointing device similar to a pen. It is used to select a displayed menu item or draw pictures on the monitor screen. It consists of a photocell and an optical system placed in a small tube.

When the tip of a light pen is moved over the monitor screen and the pen button is pressed, its photocell sensing element detects the screen location and sends the corresponding signal to the CPU.

Track Ball

Track ball is an input device that is mostly used in notebook or laptop computer, instead of a mouse. This is a ball which is half inserted and by moving fingers on the ball, the pointer can be moved.

Since the whole device is not moved, a track ball requires less space than a mouse. A track ball comes in various shapes like a ball, a button, or a square.

Scanner

Scanner is an input device, which works more like a photocopy machine. It is used when some information is available on paper and it is to be transferred to the hard disk of the computer for further manipulation.

Scanner captures images from the source which are then converted into a digital form that can be stored on the disk. These images can be edited before they are printed.

Digitizer

Digitizer is an input device which converts analog information into digital form. Digitizer can convert a signal from the television or camera into a series of numbers that could be stored in a computer. They can be used by the computer to create a picture of whatever the camera had been pointed at.

Digitizer is also known as Tablet or Graphics Tablet as it converts graphics and pictorial data into binary inputs. A graphic tablet as digitizer is used for fine works of drawing and image manipulation applications.

Microphone

Microphone is an input device to input sound that is then stored in a digital form.

The microphone is used for various applications such as adding sound to a multimedia presentation or for mixing music.

Magnetic Ink Card Reader (MICR)

MICR input device is generally used in banks as there are large number of cheques to be processed every day. The bank's code number and cheque number are printed on the cheques with a special type of ink that contains particles of magnetic material that are machine readable.

This reading process is called Magnetic Ink Character Recognition (MICR). The main advantages of MICR is that it is fast and less error prone.

Optical Character Reader (OCR)

OCR is an input device used to read a printed text.

OCR scans the text optically, character by character, converts them into a machine readable code, and stores the text on the system memory.

Bar Code Readers

Bar Code Reader is a device used for reading bar coded data (data in the form of light and dark lines). Bar coded data is generally used in labelling goods, numbering the books, etc. It may be a handheld scanner or may be embedded in a stationary scanner.

Bar Code Reader scans a bar code image, converts it into an alphanumeric value, which is then fed to the computer that the bar code reader is connected to.

Optical Mark Reader (OMR)

OMR is a special type of optical scanner used to recognize the type of mark made by pen or pencil. It is used where one out of a few alternatives is to be selected and marked.

It is specially used for checking the answer sheets of examinations having multiple choice questions.

Image Scanners

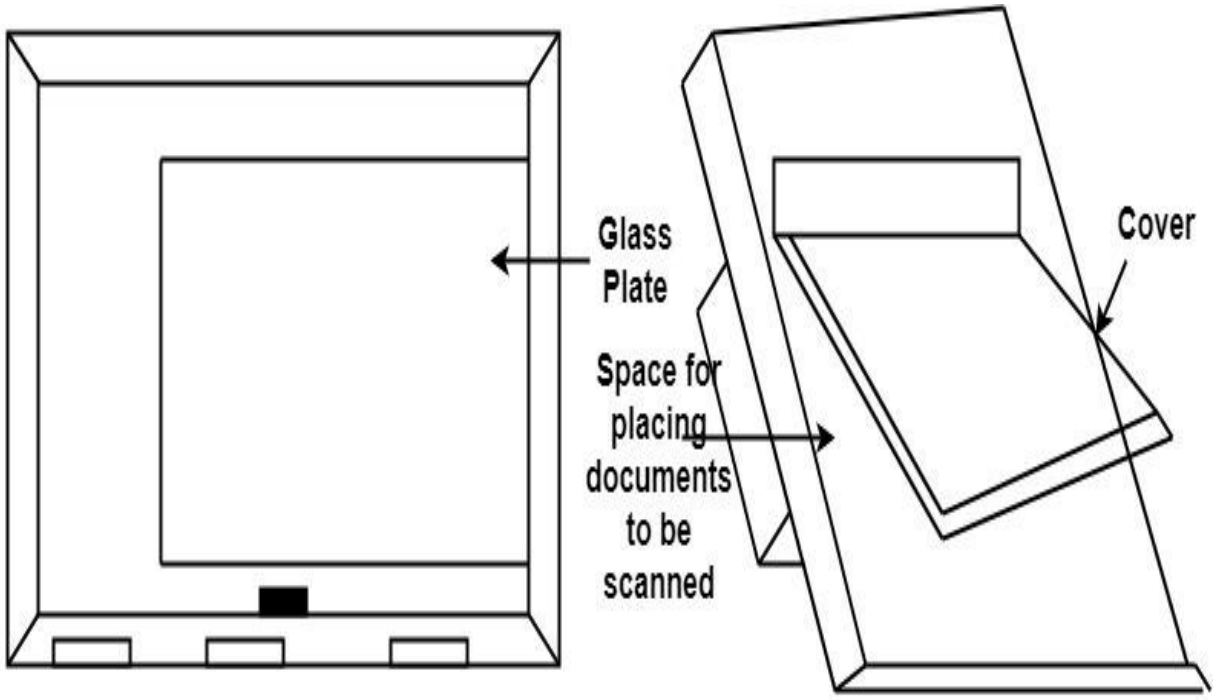
It is an input device. The data or text is written on paper. The paper is feeded to scanner. The paper written information is converted into electronic format; this format is stored in the computer. The input documents can contain text, handwritten material, picture extra.

By storing the document in a computer document became safe for longer period of time. The document will be permanently stored for the future. We can change the document when we need. The document can be printed when needed.

Scanning can be of the black and white or colored picture. On stored picture 2D or 3D rotations, scaling and other operations can be applied.

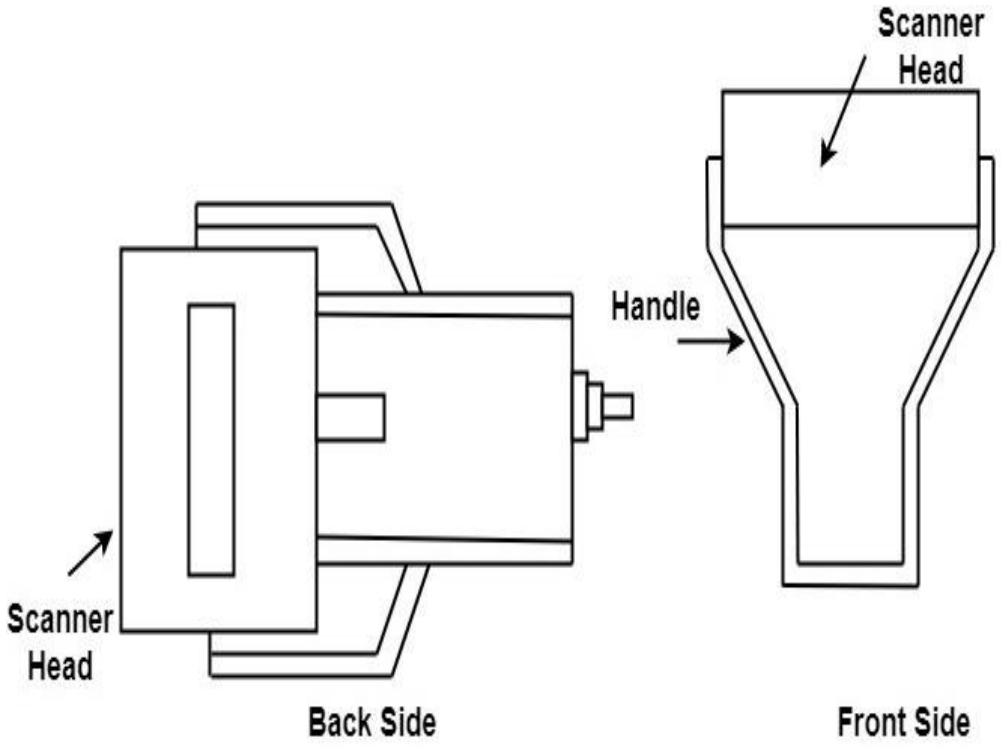
Types of image Scanner:

1. Flat Bed Scanner: It resembles a photocopy machine. It has a glass top on its top. Glass top in further covered using a lid. The document to be scanned is kept on glass plate. The light is passed underneath side of glass plate. The light is moved left to right. The scanning is done the line by line. The process is repeated until the complete line is scanned. Within 20-25 seconds a document of 4" * 6" can be scanned.



Flat Bed Scanner

2. Hand Held Scanner: It has a number of LED's (Light Emitting Diodes) the LED's are arranged in the small case. It is called a Hand held Scanner because it can be kept in hand which performs scanning. For scanning the scanner is moved over document from the top towards the bottom. Its light is on, while we move it on document. It is dragged very slowly over document. If dragging of the scanner over the document is not proper, the conversion will not correct.



Hand Held Scanner

Working exposure on graphics tools like Dream Weaver, 3D Effects etc

Adobe Dreamweaver is a software for designing web pages. These HTML web pages are fully featured. This software provides a WYSIWYG i.e., What You See Is What You Get interface for creating and editing the web pages. The Adobe Dreamweaver software supports many markup languages like HTML, XML, CSS, and JavaScript. It supports English, Spanish, French, German, Japanese, Chinese, etc....

The Dreamweaver was developed by Macromedia, and it was published in 1997. In 2005 the Adobe had purchased Dreamweaver and name it as Adobe Dreamweaver.

Features of Adobe Dreamweaver

1. Fast, flexible coding.

Creating, coding, and managing the websites becomes very easy because of the simplified coding engine. Using this software, HTML, CSS, and other web standards can be quickly learned. It speeds up the development of the web site.

2. Setup to site up in fewer steps.

It becomes very easy to set up a web site, and starter templates can be run very fast. Templates can be customized for email, About pages, blogs, e-commerce pages, newsletters, and portfolios. For reading codes, code coloring and visual hints can be used for quickly editing and updating.

3. Dynamic display on every device.

Using Adobe Dreamweaver responsive websites can build that can be fit into any screen size. This helps in previewing sites and editing makes sure that the page looks and works the same way that you want.

Below are some of the latest updates

1. Multi-monitor support for Windows

The workspace can be expended by displaying web pages on multiple monitors.

2. CEF integration

The latest version of the Chromium Embedded Framework is integrated with Dreamweaver. So now, web pages can be built using HTML5 websites and elements can be display, CSS grids, and many more.

3. Redesigned, modern UI

For customization of workspace Streamlined and the clutter-free interface can be used. It displays the tools which need to be coded.

4. Git support

Collaboration is very easy using Git support. From the Git panel, all the source code can be managed within Dreamweaver, and all the common operations can be performed

New features in Adobe Dreamweaver CC 2019

1. CEF updates

The latest version of the Chromium Embedded Framework is integrated with Dreamweaver. Designers and developers can build Websites and display Flexbox elements, CSS grids and many more.

2. ES6 support

For a quick listing of classes, methods, arrow function and generated functions, EcmaScript 6 is supported in Dreamweaver. ES6 code can be used for working with the latest JavaScript updates.

3. JavaScript refactoring

JavaScript codes can be organized using features like rename and refactor.

Following are the Pros of Adobe Dreamweaver

1. For quick scanning, it can highlight the code.

It helps in determining where is CSS, PHP, JavaScript, or HTML code is to be placed. For creating a dynamic page, this feature is very useful.

2. Helps beginners to understand the coding for a website.

The color-coding feature of highlighting helps beginners very much as it can display what a specific command can do for template and pages. By this, the learning process becomes very easy.

3. code suggestions for the user.

It is a very important benefit for beginners while coding. If the user is confused about what to do with an image, a font, or a color, then the Adobe Dreamweaver will auto-fill the suggestions directly with a drop-down menu in the code. Select your choice, which suits your websites and create a code. It is very easy.

4. User needs to switch screens.

The real-time results of the program can be seen by the users. Many programs do not allow this feature of viewing page in real-time on the same screen. In Adobo Dreamweaver coding and the designing view is on the same screen, on top is the designing view and on the bottom of the coding area.

5. The code of the developer is instantly checked.

For a beginner, it is very important to build a correct and valid in the starting phase. When coding is done in a traditional manner, then accessibility issues and mistakes in

the code can occur. Adobe Dreamweaver highlights the mistakes and shows all the issues and also check all accessibility.

6. Using word processing variations in content creation becomes easy.

In Microsoft Office, if you want to bold a text, then you just press Ctrl+B. In Adobe Dreamweaver for bolding a text, you need to highlight the text and then a word processor heads-up display which helps in creating fast changes in the highlighted section.

7. Finding and replacing items becomes easy in the created site.

In Adobe Dreamweaver, a user can find and replace items from there content, coding, or in any specific tags using find and replace tool. If any new plugin releases can be updated the code very easily. Thousands of pages can be updated in just a second using this feature.

8. A developer can tab via files like someone tabs via internet sites.

The functionality included in Dreamweaver makes the switching between files makes it very easy. Visiting multiple menus using this feature file can be kept ready. This is very beneficial in templates and also in overall design plans.

Following are the cons of Adobe Dreamweaver

1. It is not a browser-based IDE.

The output of the coding in Adobe Dreamweaver can be different in the browser because the browser will interpret the code in a different way. This becomes a huge disadvantage for the developers because it will not respond in cross-platform.

2. A lot of time is required to learn the interface.

When you first start with the adobe Dreamweaver, then you will get dozen of items for accessing. Depending on the file that you will open, 50+ different things can be seen on the screen. For a beginner, it would be very intimidating to start the coding.

3. What you are seeing is not necessary that you will always get.

This is a very disadvantage because here design view proves that it is not a browser-based view. For a developer, it becomes difficult for the positioning of items. It is 100% guaranteed that the output will be very disappointed in the browser.

4. There is no specific automatic coding option.

The Adobe Dreamweaver was designed to meet the requirements of the users. Because of this, the code becomes very lengthy. The validation of one line takes 15-20 lines of code in adobe Dreamweaver.

5. Global styling is a major issue.

If a developer wants to use the properties bar on the text within the old version of Adobe Dreamweaver, then it will add an undefined and not styled document to the coding. In case if you want many dozens of undefined styles, then it will create an issue for styling the website.

6. Pressing enter after completion of a line affects the size.

Generally, when we type a line after completion of it, we press the enter key from the keyboard. In Adobe Dreamweaver, when you are writing a line, and after completion of it, you press the enter key, then Dreamweaver will treat it as a paragraph. Also, the text will appear around the image.

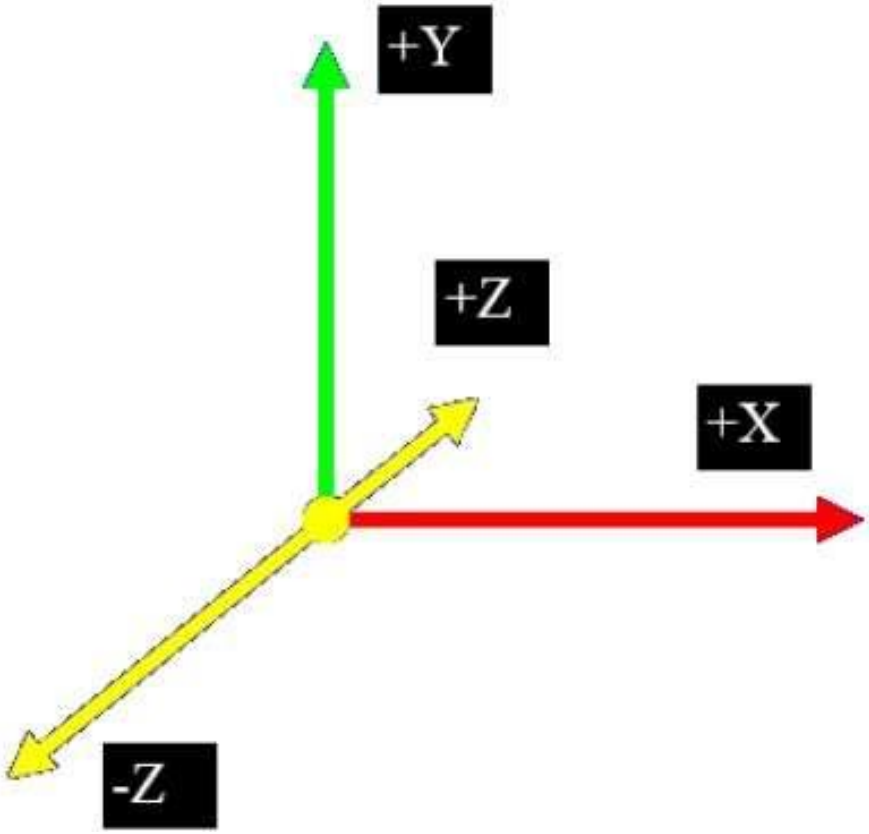
7. A lot of additional features by which a user is not familiar.

As adobe Dreamweaver is designed for the user to make there work easy, but most of the users find that a lot of features are present which they have never used.

3D Effects

In the 2D system, we use only two coordinates X and Y but in 3D, an extra coordinate Z is added. 3D graphics techniques and their application are fundamental to the entertainment, games, and computer-aided design industries. It is a continuing area of research in scientific visualization.

Furthermore, 3D graphics components are now a part of almost every personal computer and, although traditionally intended for graphics-intensive software such as games, they are increasingly being used by other applications.

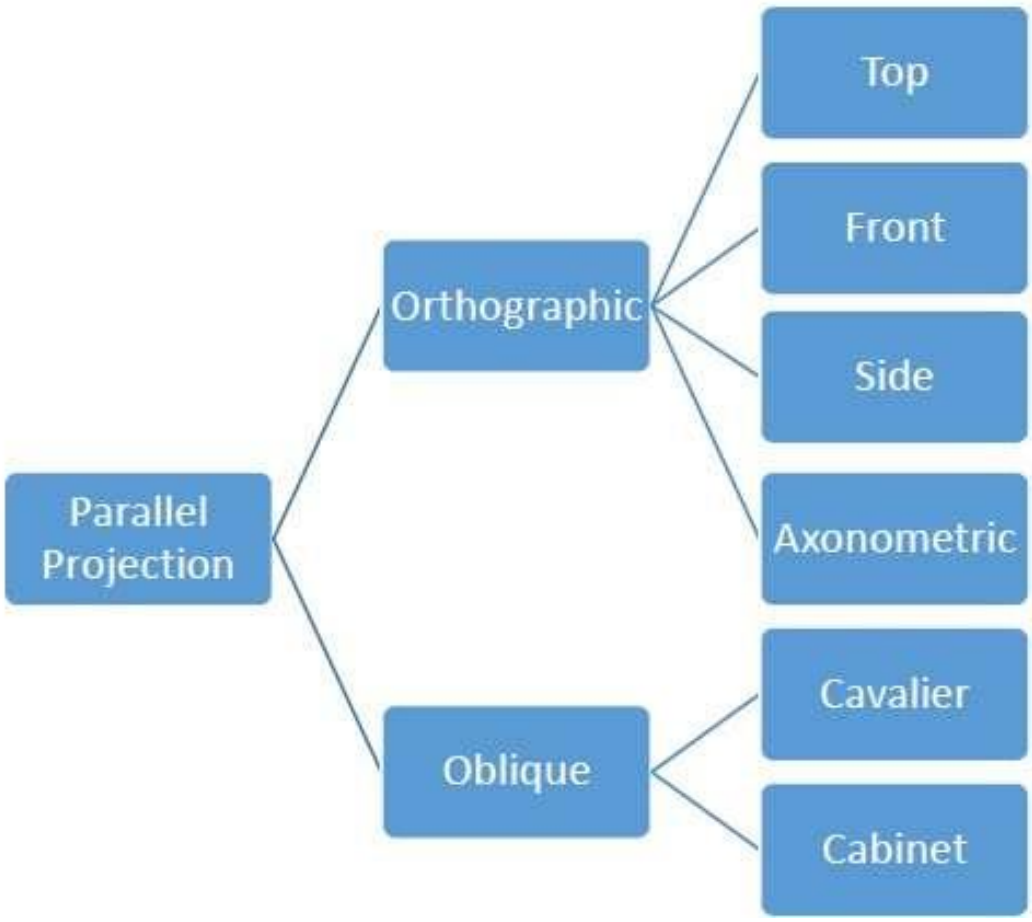


Parallel Projection

Parallel projection discards z-coordinate and parallel lines from each vertex on the object are extended until they intersect the view plane. In parallel projection, we specify a direction of projection instead of center of projection.

In parallel projection, the distance from the center of projection to project plane is infinite. In this type of projection, we connect the projected vertices by line segments which correspond to connections on the original object.

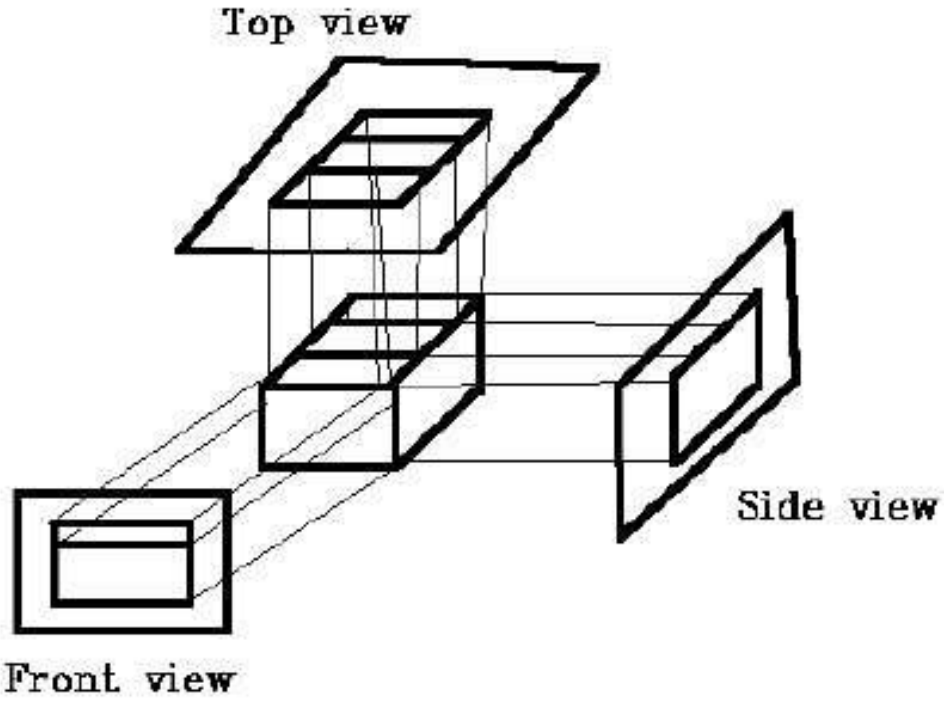
Parallel projections are less realistic, but they are good for exact measurements. In this type of projections, parallel lines remain parallel and angles are not preserved. Various types of parallel projections are shown in the following hierarchy.



Orthographic Projection

In orthographic projection the direction of projection is normal to the projection of the plane. There are three types of orthographic projections −

- Front Projection
- Top Projection
- Side Projection

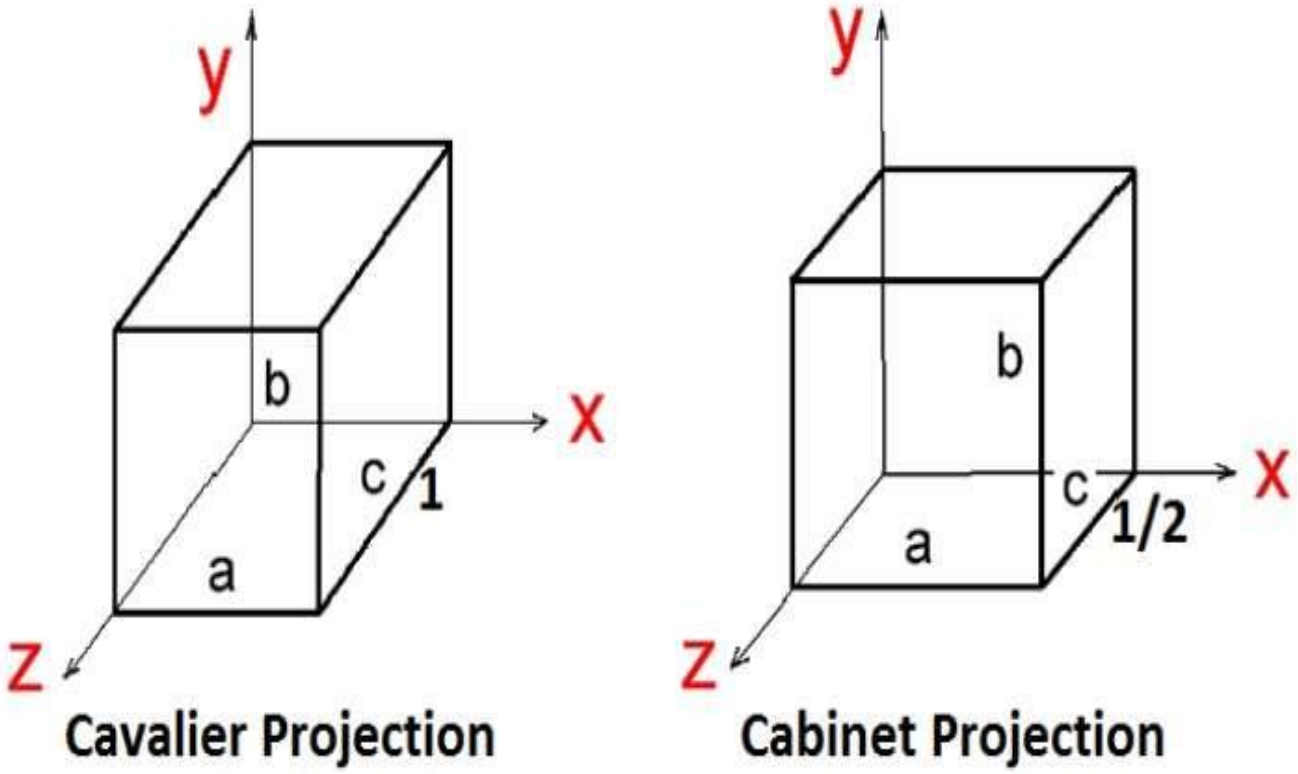


Oblique Projection

In oblique projection, the direction of projection is not normal to the projection of plane. In oblique projection, we can view the object better than orthographic projection.

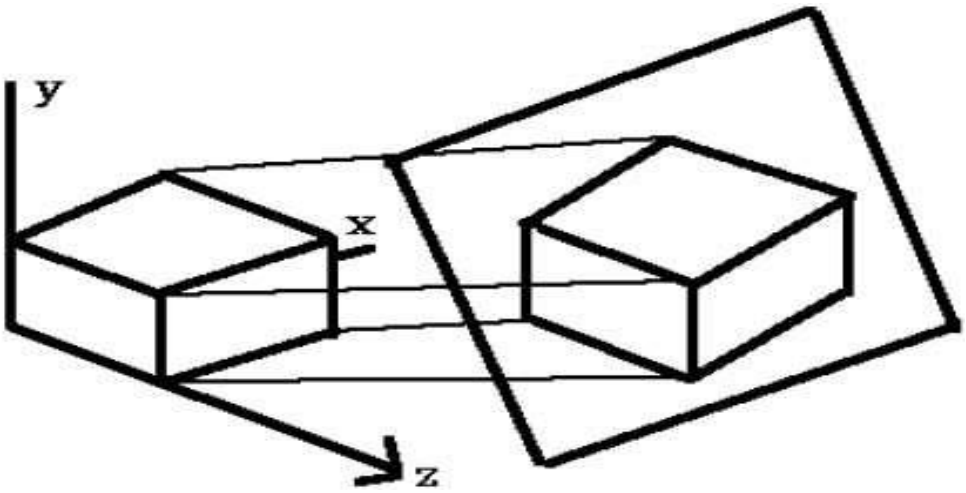
There are two types of oblique projections − Cavalier and Cabinet. The Cavalier projection makes 45° angle with the projection plane. The projection of a line perpendicular to the view plane has the same length as the line itself in Cavalier projection. In a cavalier projection, the foreshortening factors for all three principal directions are equal.

The Cabinet projection makes 63.4° angle with the projection plane. In Cabinet projection, lines perpendicular to the viewing surface are projected at ½ their actual length. Both the projections are shown in the following figure −



Isometric Projections

Orthographic projections that show more than one side of an object are called axonometric orthographic projections. The most common axonometric projection is an isometric projection where the projection plane intersects each coordinate axis in the model coordinate system at an equal distance. In this projection parallelism of lines are preserved but angles are not preserved. The following figure shows isometric projection −

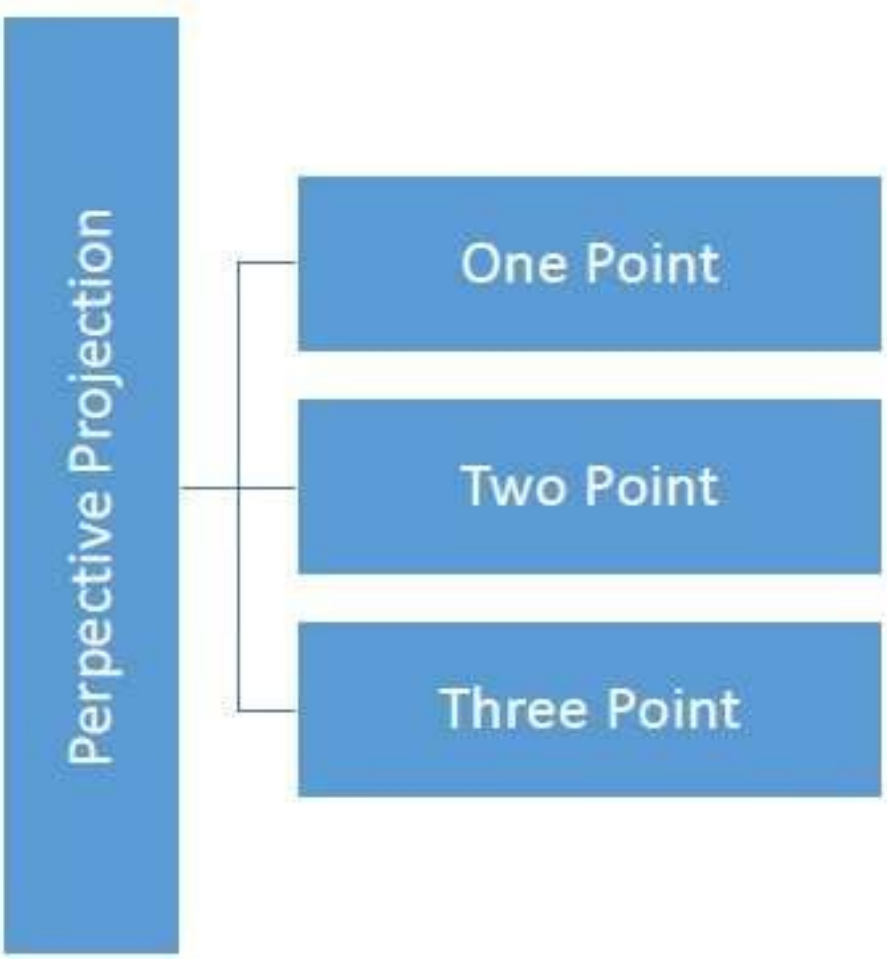


Perspective Projection

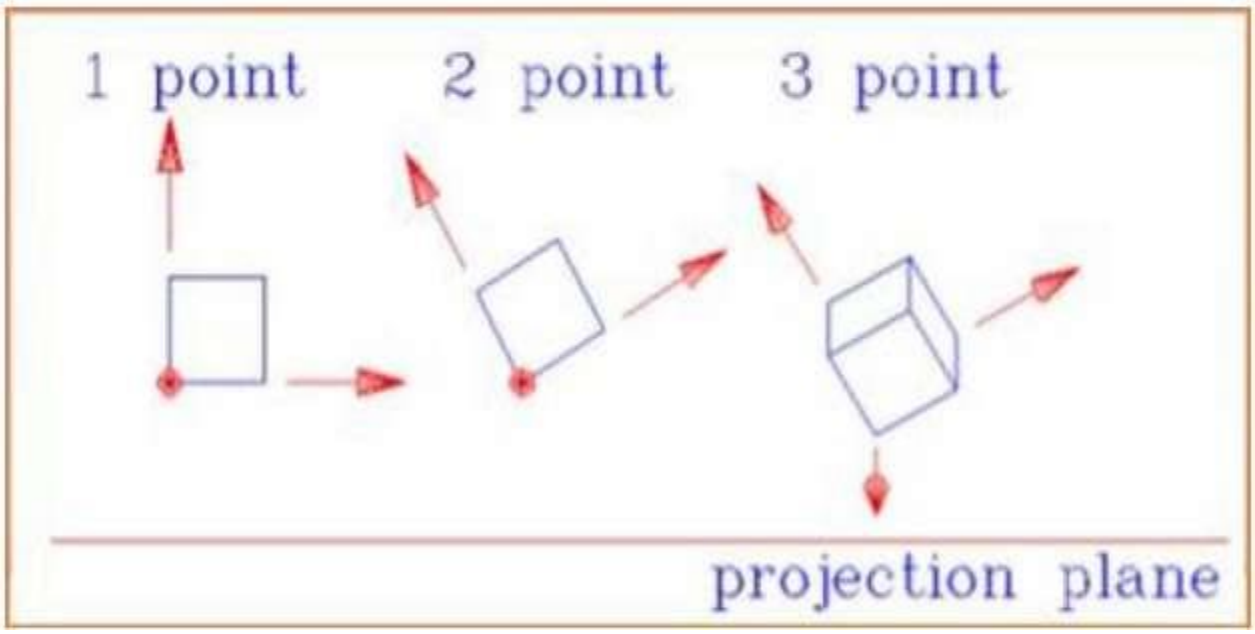
In perspective projection, the distance from the center of projection to project plane is finite and the size of the object varies inversely with distance which looks more realistic.

The distance and angles are not preserved and parallel lines do not remain parallel. Instead, they all converge at a single point called center of projection or projection reference point. There are 3 types of perspective projections which are shown in the following chart.

- **One point** perspective projection is simple to draw.
- **Two point** perspective projection gives better impression of depth.
- **Three point** perspective projection is most difficult to draw.



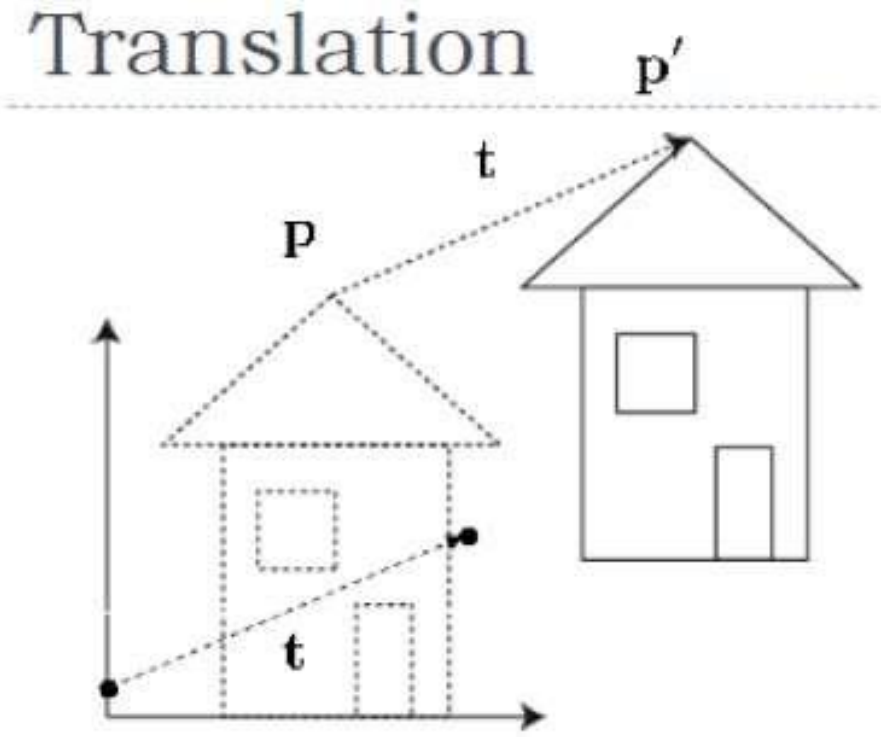
The following figure shows all the three types of perspective projection −



Translation

In 3D translation, we transfer the Z coordinate along with the X and Y coordinates. The process for translation in 3D is similar to 2D translation. A translation moves an object into a different position on the screen.

The following figure shows the effect of translation −



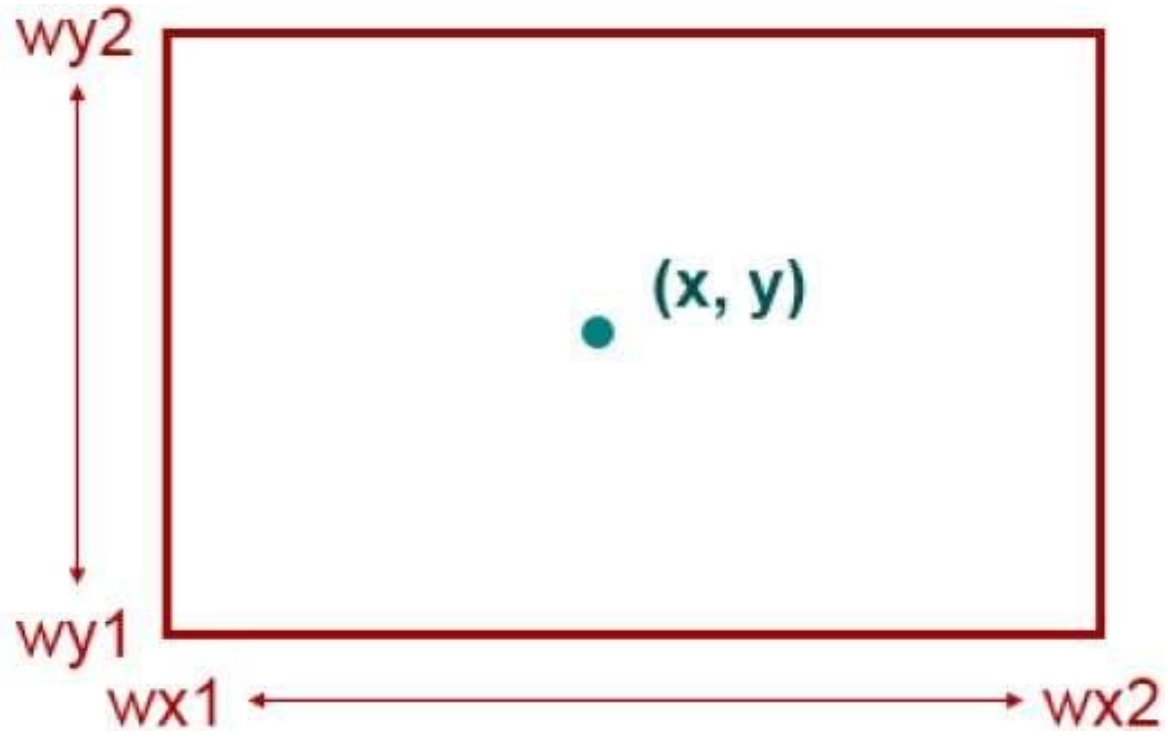
Clipping

The primary use of clipping in computer graphics is to remove objects, lines, or line segments that are outside the viewing pane. The viewing transformation is insensitive to the position of points relative to the viewing volume − especially those points behind the viewer − and it is necessary to remove these points before generating the view.

Point Clipping

Clipping a point from a given window is very easy. Consider the following figure, where the rectangle indicates the window. Point clipping tells us whether the given point X,YX,Y is within the given window or not; and decides whether we will use the minimum and maximum coordinates of the window.

The X-coordinate of the given point is inside the window, if X lies in between Wx1 ≤ X ≤ Wx2. Same way, Y coordinate of the given point is inside the window, if Y lies in between Wy1 ≤ Y ≤ Wy2.

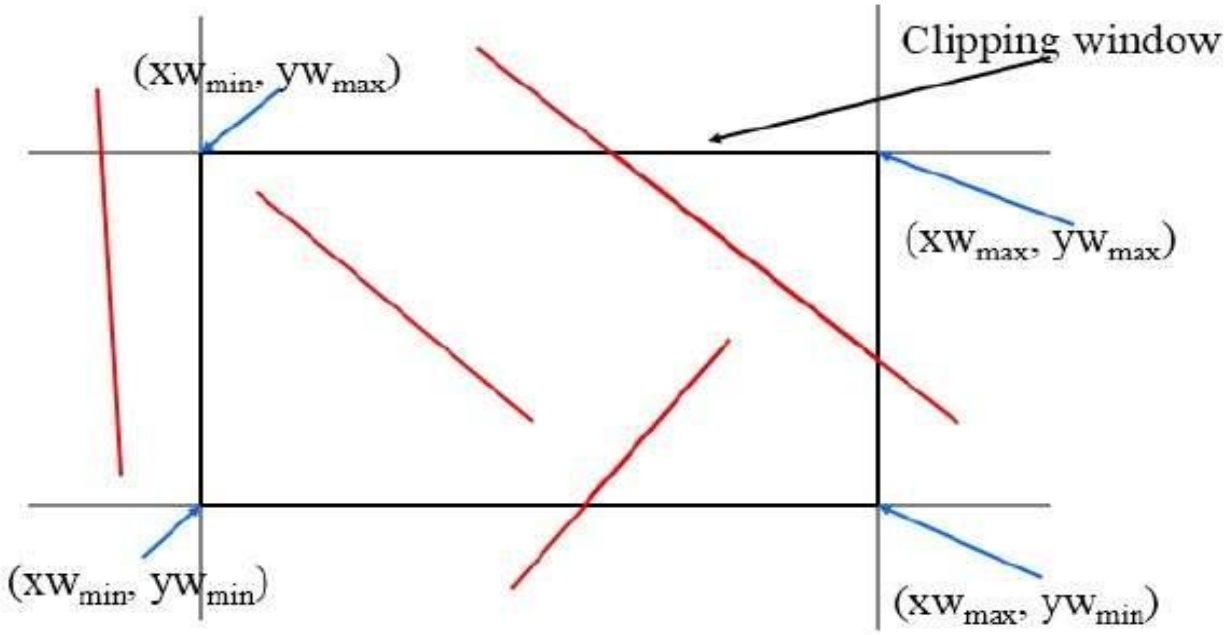


Line Clipping

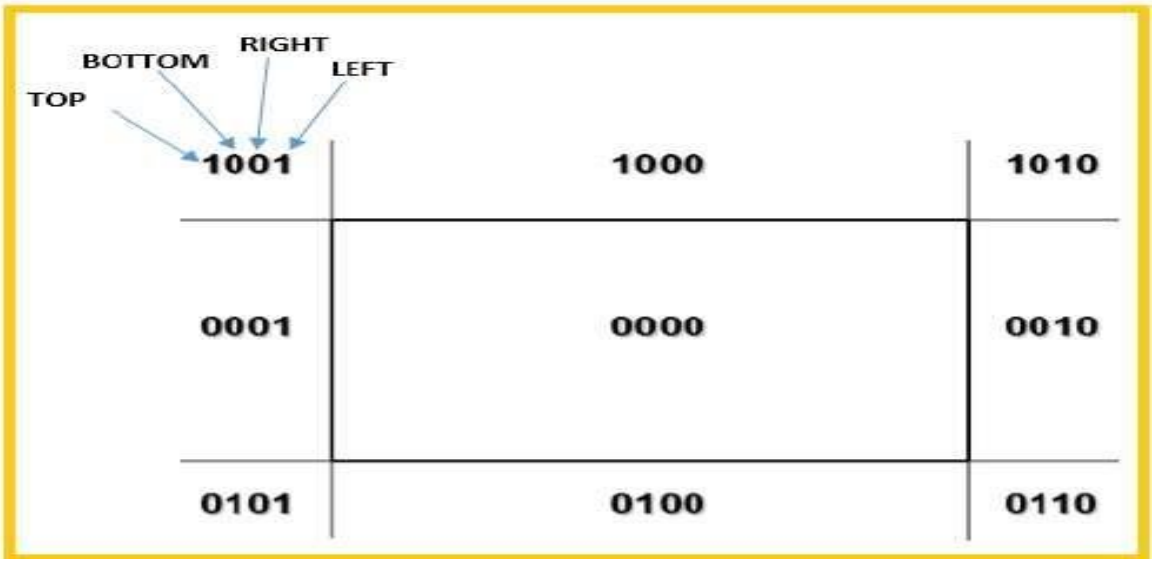
The concept of line clipping is same as point clipping. In line clipping, we will cut the portion of line which is outside of window and keep only the portion that is inside the window.

Cohen-Sutherland Line Clippings

This algorithm uses the clipping window as shown in the following figure. The minimum coordinate for the clipping region is (XWmin,YWmin)(XWmin,YWmin) and the maximum coordinate for the clipping region is (XWmax,YWmax)(XWmax,YWmax).



We will use 4-bits to divide the entire region. These 4 bits represent the Top, Bottom, Right, and Left of the region as shown in the following figure. Here,
the TOP and LEFT bit is set to 1 because it is the TOP-LEFT corner.



There are 3 possibilities for the line −

- Line can be completely inside the window ThislineshouldbeacceptedThislineshouldbeaccepted.

- Line can be completely outside of the window ThislinewillbecompletelyremovedfromtheregionThislinewillbecompletelyremovedfromtheregion.

- Line can be partially inside the window WewillfindintersectionpointanddrawonlythatportionoflinethatisinsideregionWewillfindintersectionpointanddrawonlythatportionoflinethatisinsideregion.

Algorithm

Step 1 − Assign a region code for each endpoints.

Step 2 − If both endpoints have a region code 0000 then accept this line.

Step 3 − Else, perform the logical ANDoperation for both region codes.

Step 3.1 − If the result is not 0000**,** then reject the line.

Step 3.2 − Else you need clipping.

Step 3.2.1 − Choose an endpoint of the line that is outside the window.

Step 3.2.2 − Find the intersection point at the window boundary baseonregioncodebaseonregioncode.

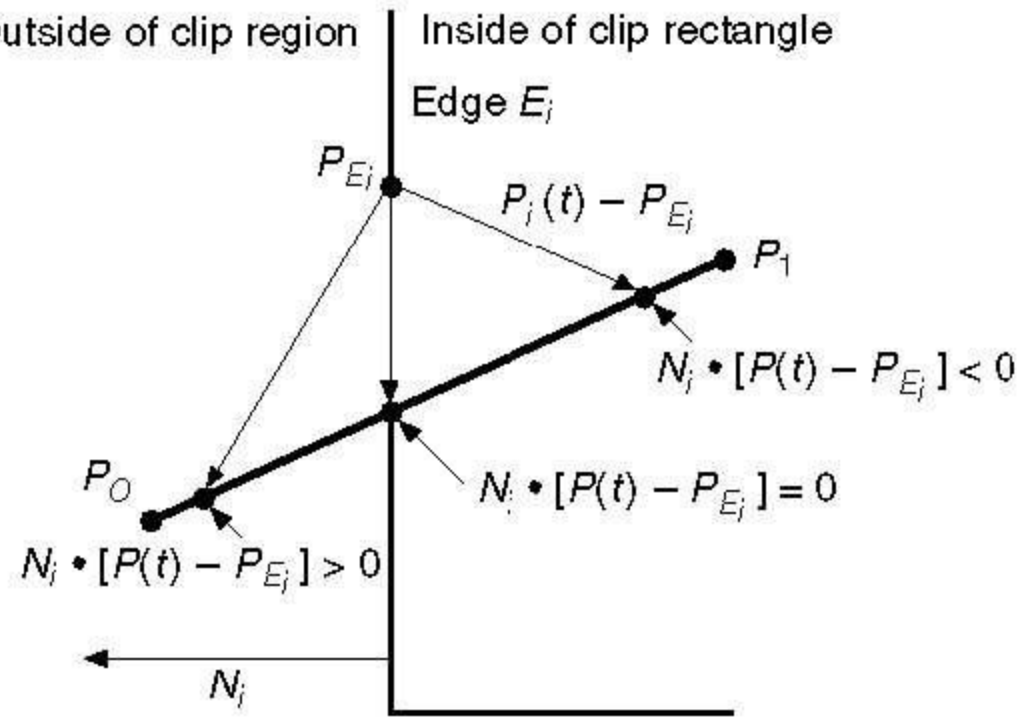
Step 3.2.3 − Replace endpoint with the intersection point and update the region code.

Step 3.2.4 − Repeat step 2 until we find a clipped line either trivially accepted or trivially rejected.

Step 4 − Repeat step 1 for other lines.

Cyrus-Beck Line Clipping Algorithm

This algorithm is more efficient than Cohen-Sutherland algorithm. It employs parametric line representation and simple dot products.



Parametric equation of line is −

P_0P_1:$P(t) = P_0 + t(P_1 - P_0)$

Let N_i be the outward normal edge E_i. Now pick any arbitrary point P_{Ei} on edge E_i then the dot product $N_i.[Ptt – P_{Ei}]$ determines whether the point Ptt is “inside the clip edge” or “outside” the clip edge or “on” the clip edge.

The point Ptt is inside if $N_i.[Ptt – P_{Ei}] < 0$
The point Ptt is outside if $N_i.[Ptt – P_{Ei}] > 0$

The point Ptt is on the edge if $N_i.[Ptt – P_{Ei}] = 0$ IntersectionpointIntersectionpoint

$N_i.[Ptt – P_{Ei}] = 0$
$N_i.[P_0 + t(P_1 - P_0) – P_{Ei}] = 0$ ReplacingP(tReplacingP(t with $P_0 + t(P_1 - P_0)$)

$N_i.[P_0 – P_{Ei}] + N_i.t[P_1 - P_0] = 0$

$N_i.[P_0 – P_{Ei}] + N_i \cdot tD = 0$ (substituting D for $[P_1 - P_0]$)

$N_i.[P_0 – P_{Ei}] = - N_i \cdot tD$

The equation for t becomes,

t=Ni.[Po−PEi]−Ni.Dt=Ni.[Po−PEi]−Ni.D

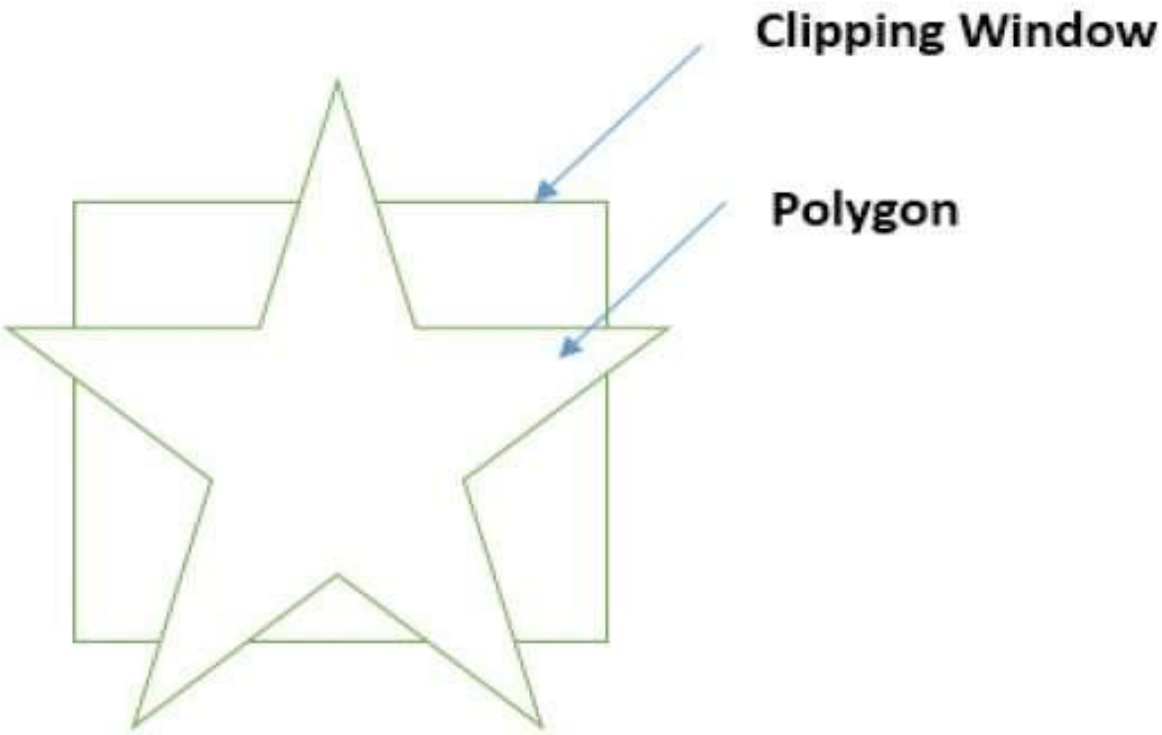
It is valid for the following conditions −

- $N_i \neq 0$ errorcannothappenerrorcannothappen
- $D \neq 0$ ($P_1 \neq P_0$)
- $N_i \cdot D \neq 0$ (P_0P_1 not parallel to E_i)

Polygon Clipping SutherlandHodgmanAlgorithmSutherlandHodgmanAlgorithm

A polygon can also be clipped by specifying the clipping window. Sutherland Hodgeman polygon clipping algorithm is used for polygon clipping. In this algorithm, all the vertices of the polygon are clipped against each edge of the clipping window.

First the polygon is clipped against the left edge of the polygon window to get new vertices of the polygon. These new vertices are used to clip the polygon against right edge, top edge, bottom edge, of the clipping window as shown in the following figure.



While processing an edge of a polygon with clipping window, an intersection point is found if edge is not completely inside clipping window and the a partial edge from the intersection point to the outside edge is clipped. The following figures show left, right, top and bottom edge clippings −

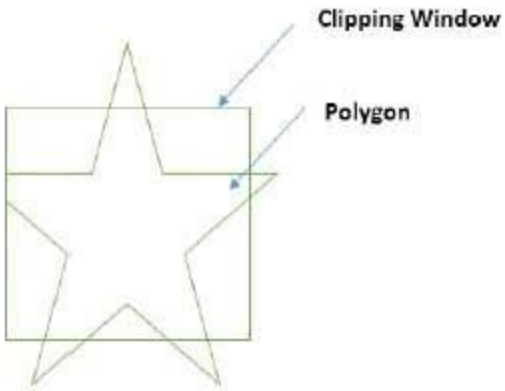


Figure: Clipping Left Edge

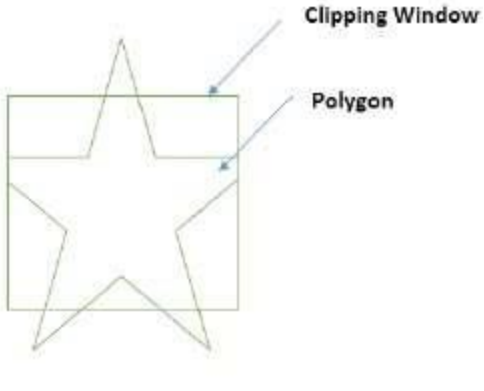


Figure: Clipping Right Edge

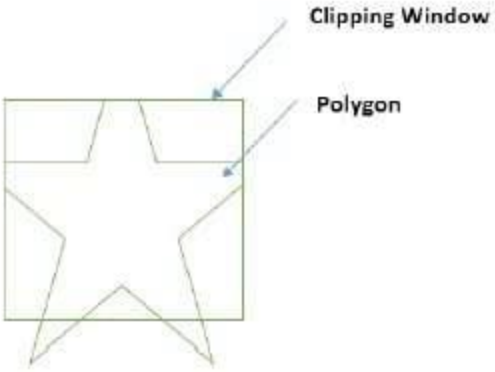


Figure: Clipping Top Edge

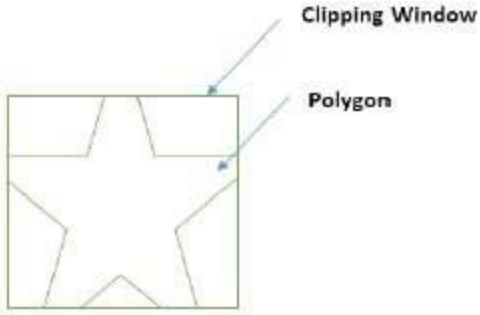


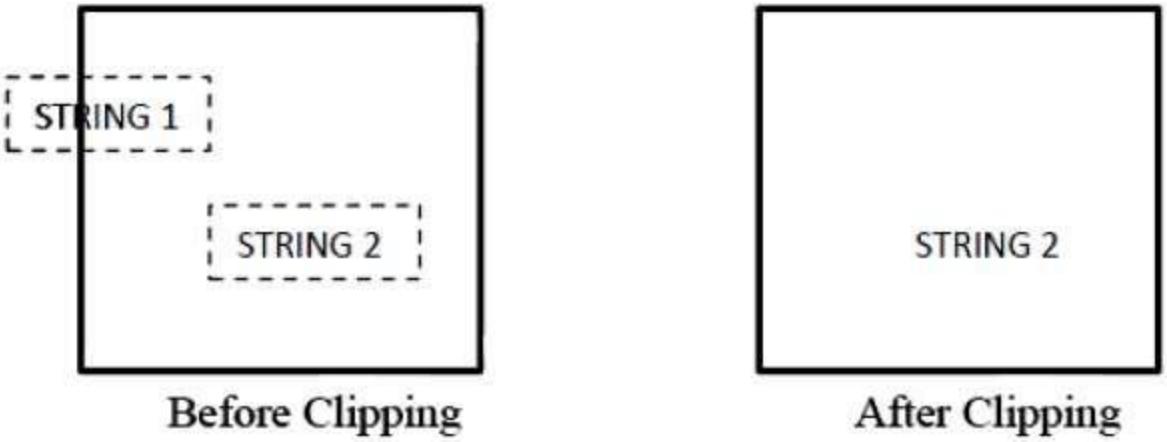
Figure: Clipping Bottom Edge

Text Clipping

Various techniques are used to provide text clipping in a computer graphics. It depends on the methods used to generate characters and the requirements of a particular application. There are three methods for text clipping which are listed below −

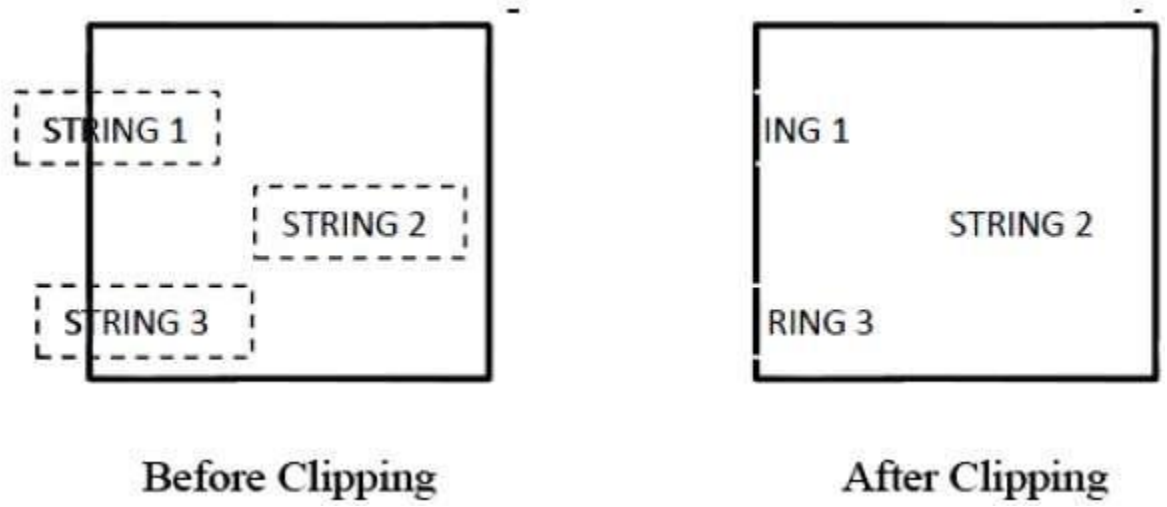
- All or none string clipping
- All or none character clipping
- Text clipping

The following figure shows all or none string clipping −



In all or none string clipping method, either we keep the entire string or we reject entire string based on the clipping window. As shown in the above figure, STRING2 is entirely inside the clipping window so we keep it and STRING1 being only partially inside the window, we reject.

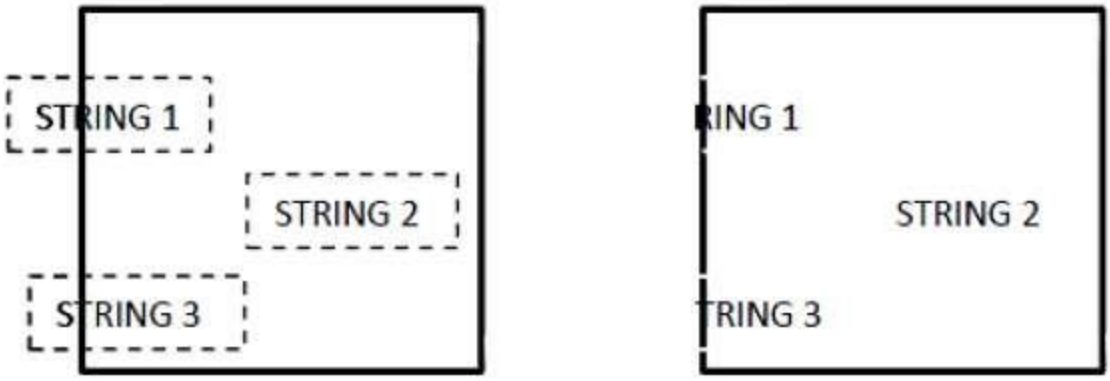
The following figure shows all or none character clipping −



This clipping method is based on characters rather than entire string. In this method if the string is entirely inside the clipping window, then we keep it. If it is partially outside the window, then −

- You reject only the portion of the string being outside
- If the character is on the boundary of the clipping window, then we discard that entire character and keep the rest string.

The following figure shows text clipping −



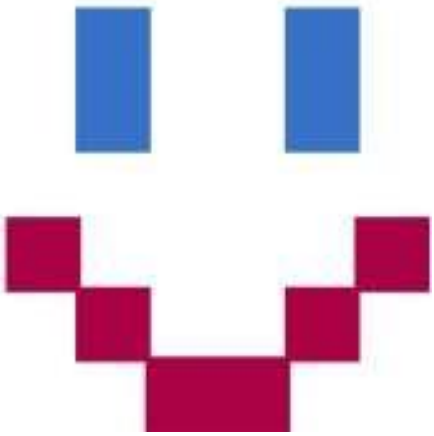
Before Clipping After Clipping

This clipping method is based on characters rather than the entire string. In this method if the string is entirely inside the clipping window, then we keep it. If it is partially outside the window, then

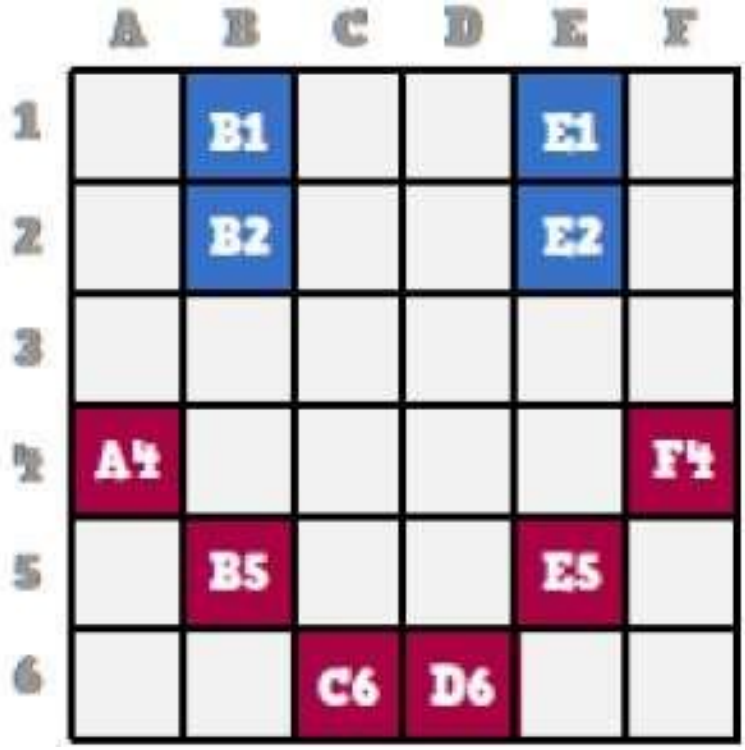
- You reject only the portion of string being outside.
- If the character is on the boundary of the clipping window, then we discard only that portion of character that is outside of the clipping window.

Bitmap Graphics

A bitmap is a collection of pixels that describes an image. It is a type of computer graphics that the computer uses to store and display pictures. In this type of graphics, images are stored bit by bit and hence it is named Bit-map graphics. For better understanding let us consider the following example where we draw a smiley face using bit-map graphics.



Now we will see how this smiley face is stored bit by bit in computer graphics.



By observing the original smiley face closely, we can see that there are two blue lines which are represented as B1, B2 and E1, E2 in the above figure.

In the same way, the smiley is represented using the combination bits of A4, B5, C6, D6, E5, and F4 respectively.

The main disadvantages of bitmap graphics are −

- We cannot resize the bitmap image. If you try to resize, the pixels get blurred.
- Colored bitmaps can be very large.

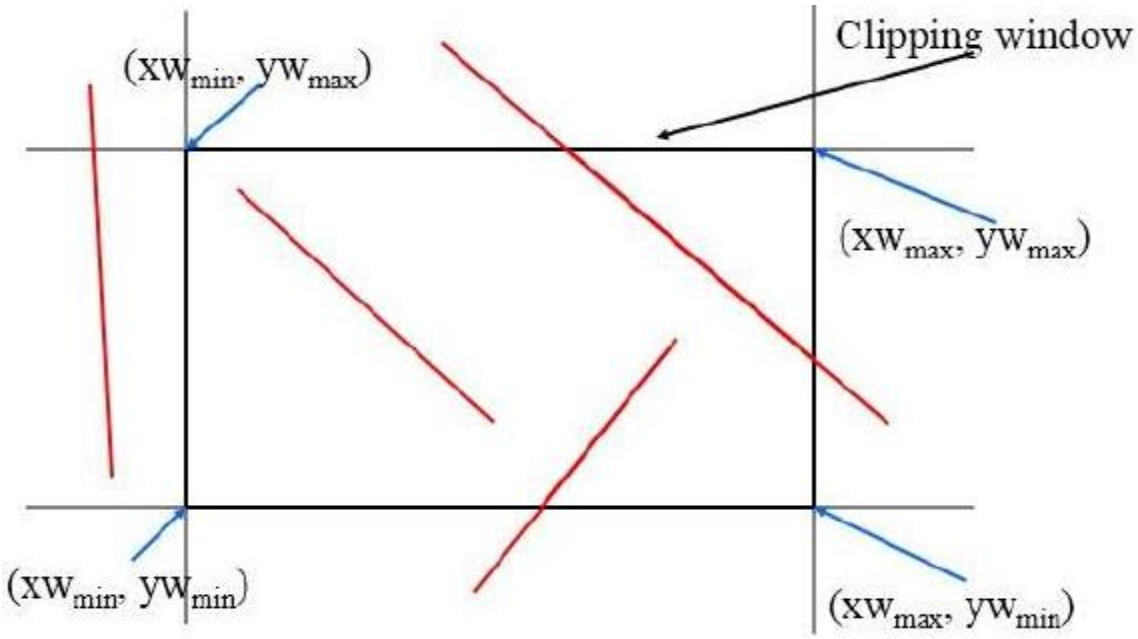
Southland
Cohen Algorithm

The concept of line clipping is same as point clipping. In line clipping, we will cut the portion of line which is outside of window and keep only the portion that is inside the window.

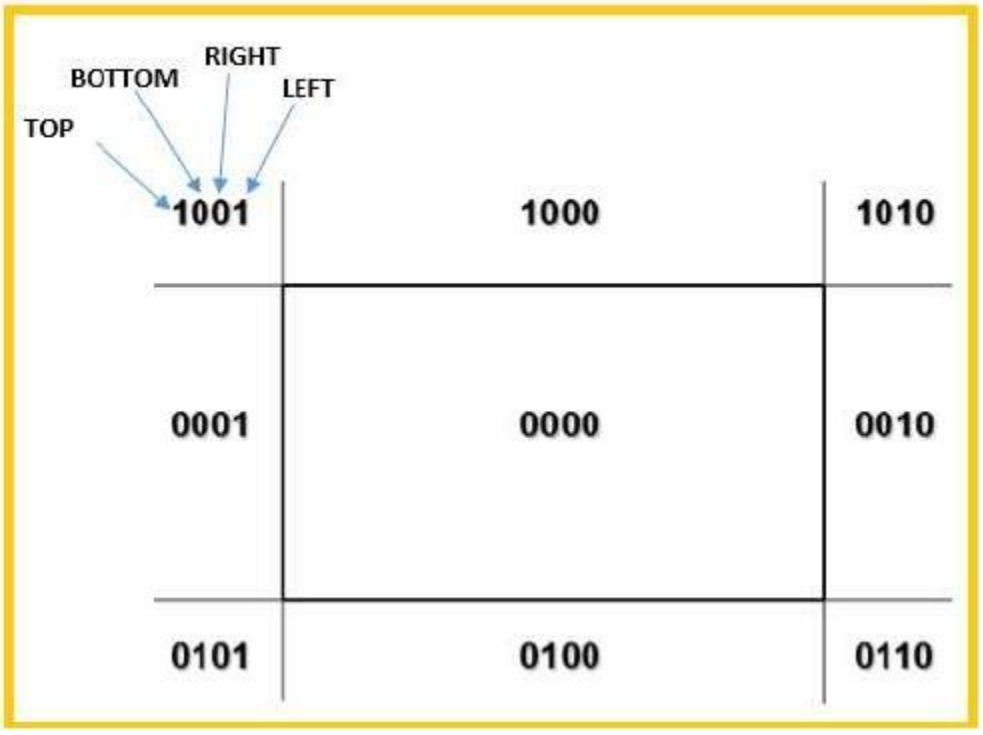
Cohen-Sutherland Line Clippings:

- This algorithm uses the clipping window as shown in the following figure. The minimum coordinate for the clipping region is(

XWmmiinn,YWmmiinn)(XWmmiinn,YWmmiinn) and the maximum coordinate for the clipping region is (XWmmaaxx,YWmmaaxx)(XWmmaaxx,YWmmaaxx).



- We will use 4-bits to divide the entire region. These 4 bits represent the Top, Bottom, Right, and Left of the region as shown in the following figure. Here, the TOP and LEFT bit is set to 1 because it is the TOP-LEFT corner.



- There are 3 possibilities for the line:
- Line can be completely inside the window (This line should be accepted).
- Line can be completely outside of the window (This line will be completely removed from the region).
- Line can be partially inside the window (We will find intersection point and draw only that portion of line that is inside region).

Algorithm:

Step 1 − Assign a region code for each endpoints.

Step 2 − If both endpoints have a region code 0000 then accept this line.

Step 3 − Else, perform the logical ANDoperation for both region codes.

Step 3.1 − If the result is not 0000, then reject the line.

Step 3.2 − Else you need clipping.

Step 3.2.1 − Choose an endpoint of the line that is outside the window.

Step 3.2.2 − Find the intersection point at the window boundary (base on region code).

Step 3.2.3 − Replace endpoint with the intersection point and update the region code.

Step 3.2.4 − Repeat step 2 until we find a clipped line either trivially accepted or trivially rejected.

Step 4 − Repeat step 1 for other lines.

Conclusion:

In summary, the C-S algorithm is efficient when outcode testing can be done cheaply (for example, by doing bitwise operations in assembly language) and trivial acceptance or rejection is applicable to the majority of line segments .(For example, large windows - everything is inside , or small windows - everything is outside).

Cyrus-Beck Algorithm

Cyrus Beck Line clipping algorithm is actually, a parametric line-clipping algorithm. The term parametric means that we require finding the value of the parameter t in the parametric representation of the line segment for the point at that the segment intersects the clipping edge. For good understanding, identify the Figure (a) as in above, here P Q is a line segment, that is intersecting at the two edges of the convex window.

Note: The algorithm is appropriate to the "convex polygonal window".

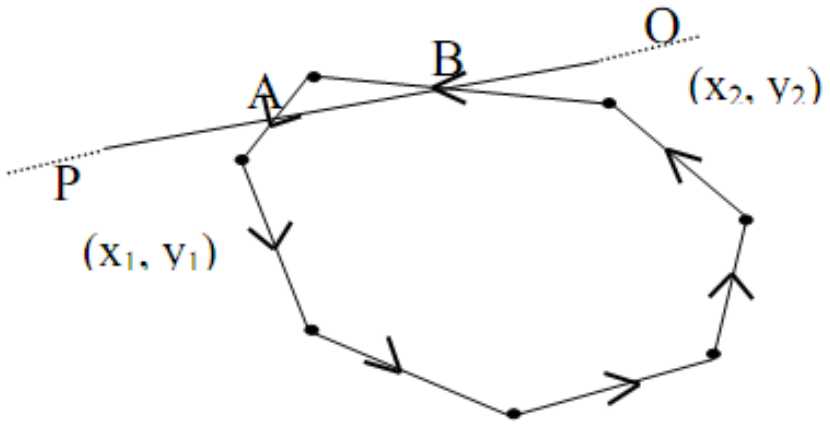


Figure: (a): Interaction of line PQ and Window

Here, just again call the parametric equation of line segment PQ that we have already studied.

This is simply $P + t (Q - P)$ $0 \leq t \leq 1$

Here, $t \rightarrow$ linear parameter incessantly changes value.

$\therefore P + t (Q - P) \Rightarrow (x_1, y_1) + t (x_2 - x_1, y_2 - y_1) = (x, y)$ be any point on PQ. ------ (1)

For such equation (1) we have subsequent cases:

1) While $t = 0$ we obtain the point P.

2) While $t = 1$ we get the point Q.

3) While t varies then $0 \leq t \leq 1$ then line in between point P and Q is traced.

For $t = ½$ we find the mid-point of PQ.

4) While $t < 0$ line on Left hand side of P is traced.

5) While $t > 1$ line on Right hand side of Q is traced.

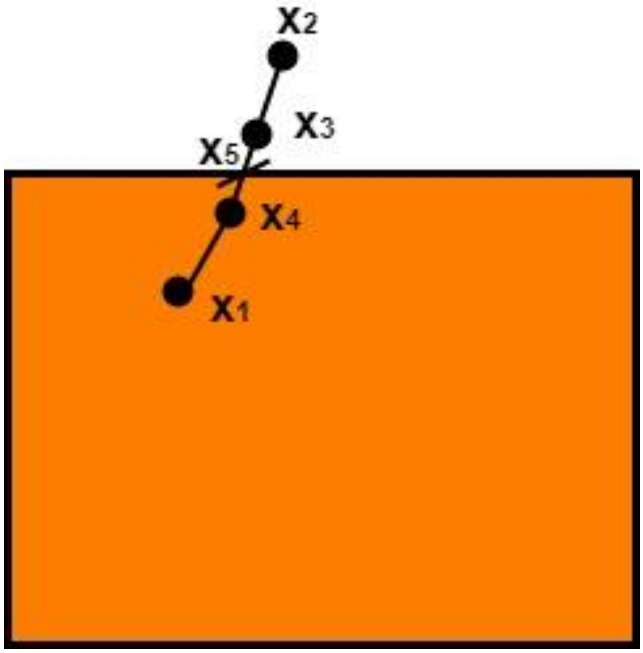
Consequently, the variation in parameter t is actually generating line in point wise method. The range of the parameter values will discover the portion to be clipped through any of convex polygonal region consisting of n-vertices or lattice points to be identified through the user. There is one type of clipping situation is shown in Figure (a) of Interaction of line PQ and Window.

Midpoint Subdivision Algorithm

It is used for clipping line. The line is divided in two parts. Mid points of line is obtained by dividing it in two short segments. Again division is done, by finding midpoint. This

process is continued until line of visible and invisible category is obtained. Let (x_i,y_i) are midpoint

$$x_m = \frac{x_1+x_2}{2} \quad y_m = \frac{y_1+y_2}{2}$$



Step1: Find $\frac{x_2+x_1}{2}$ i. e. $x_3 = \frac{x_2+x_1}{2}$

Step2: Find $x_4 = \frac{x_3+x_1}{2}$

Step3: Find $x_5 = \frac{x_4+x_3}{2}$

$x_5$lie on point of intersection of boundary of window.

Advantage of midpoint subdivision Line Clipping:

It is suitable for machines in which multiplication and division operation is not possible. Because it can be performed by introducing clipping divides in hardware.

Algorithm of midpoint subdivision Line Clipping:

Step1: Calculate the position of both endpoints of the line

Step2: Perform OR operation on both of these endpoints

Step3: If the OR operation gives 0000
then
Line is guaranteed to be visible
else
Perform AND operation on both endpoints.
If AND ≠ 0000
then the line is invisible
else
AND=6000
then the line is clipped case.

Step4: For the line to be clipped. Find midpoint
$X_m=(x_1+x_2)/2$
$Y_m=(y_1+y_2)/2$
$X_m$is midpoint of X coordinate.
$Y_m$is midpoint of Y coordinate.

Step5: Check each midpoint, whether it nearest to the boundary of a window or not.

Step6: If the line is totally visible or totally rejected not found then repeat step 1 to 5.

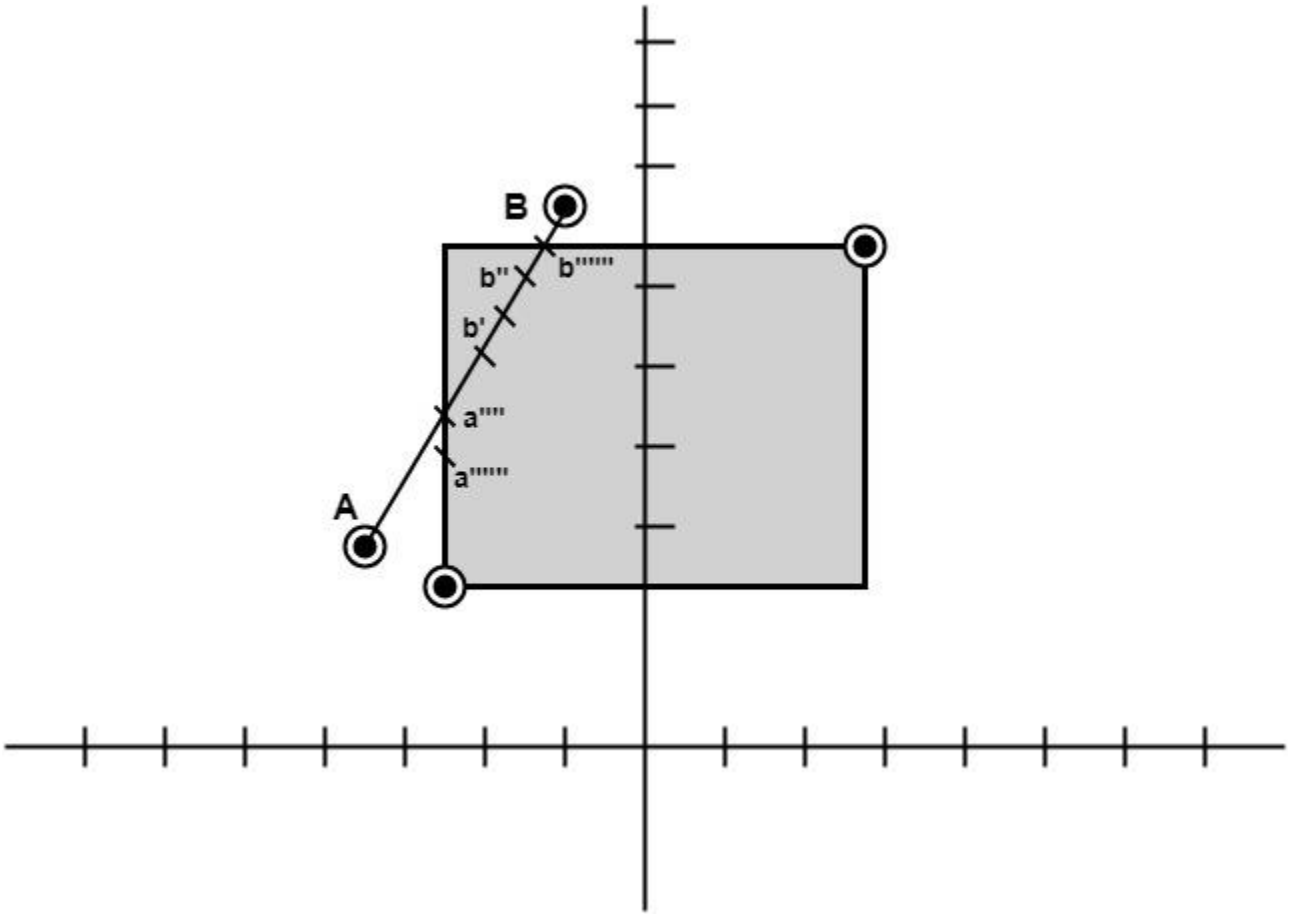
Step7: Stop algorithm.

Example: Window size is (-3, 1) to (2, 6). A line AB is given having co-ordinates of A (-4, 2) and B (-1, 7). Does this line visible. Find the visible portion of the line using midpoint subdivision?

Solution:

Step1: Fix point A (-4, 2)

$$b = \left(\frac{-4+(-1)}{2}, \frac{2+7}{2}\right) = \frac{-5}{2}, \frac{9}{2} = (-2,4)$$



Step2: Find b''=mid of b'and b

$$b'' = \left(\frac{-2+(-1)}{2}, \frac{4+7}{2}\right)$$

b''= (-1, 5)

So (-1, 5) is better than (2, 4)
Find b''&bb''(-1, 5) b (-1, 7)

$$b'''' = \left(\frac{-1+(-1)}{2}, \frac{5+7}{2}\right)$$

$$b = (-1,6)$$

So B''''to B length of line will be clipped from upper side

Now considered left-hand side portion.

A and B""are now endpoints

Find mid of A and B""

A (-4, 2) B ""(-1, 6)

$$a' = \left(\frac{-4+(-1)}{2}, \frac{2+6}{2}\right)$$

$$a' = (-2.5,4)$$

$$a' = (-2,4)$$

Now good a (-4, 2) and $a'(-2,4)$

$$a'' = \left(\frac{-4+(-2)}{2}, \frac{2+4}{2}\right)$$

$$a'' = (-3,3)$$

Now find mid of a'' and a

a""""=a (-4, 2) and $a''(-3, 3)$

$$= \left(\frac{-4+(-3)}{2}, \frac{2+3}{2}\right)$$

$$= \left(\frac{-7}{2}, \frac{5}{2}\right)$$

$$= (-3.5,2.5)$$

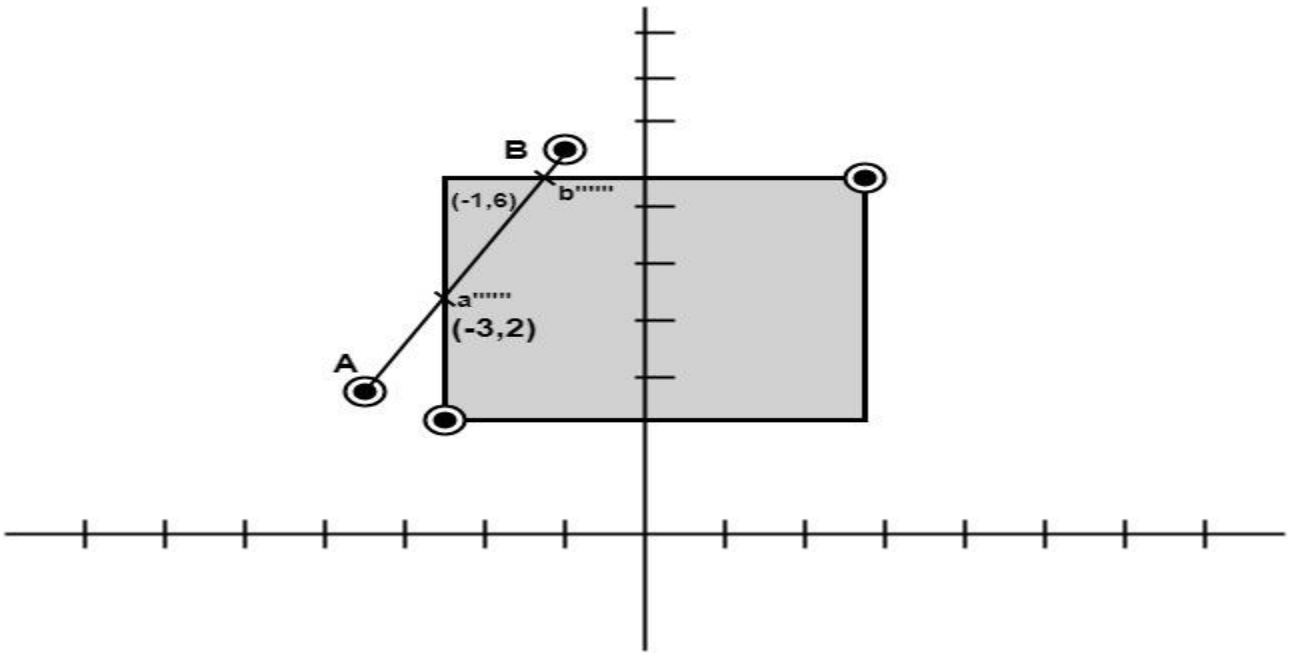
a""""= (-3, 2)

So line from A to a""""will clipped

Line after clipping from both sides will be a""to b""

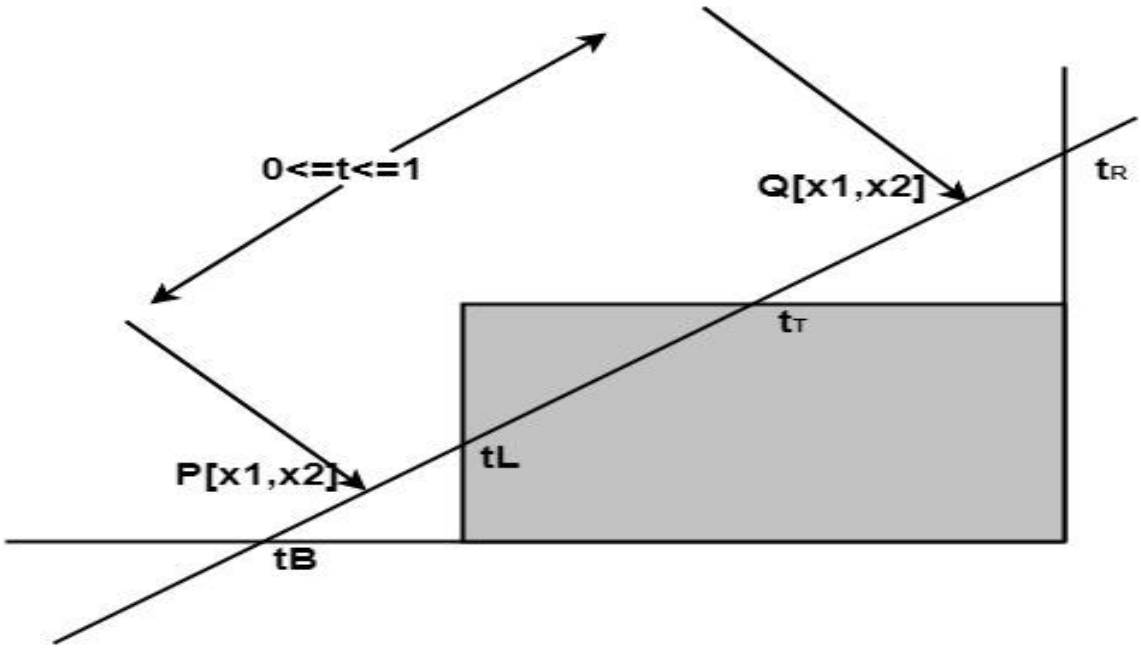
a""""= (-1, 6)

b""""= (-3, 2)



Liang-Barsky Line Clipping Algorithm:

Liang and Barsky have established an algorithm that uses floating-point arithmetic but finds the appropriate endpoints with at most four computations. This algorithm uses the parametric equations for a line and solves four inequalities to find the range of the parameter for which the line is in the viewport.



Let P(x1, y1), Q(x2, y2) is the line which we want to study. The parametric equation of the line segment from gives x-values and y-values for every point in terms of a parameter that ranges from 0 to 1. The equations are

$x=x_1+(x_2-x_1)*t=x_1+dx*t$ and $y=y_1+(y_2-y_1)*t=y_1+dy*t$

We can see that when t = 0, the point computed is P(x1, y1); and when t = 1, the point computed is Q(x2, y2).

Algorithm of Liang-Barsky Line Clipping:

1. Set $t_{min}=0$ and $t_{max}=1$

2. Calculate the values t_L,t_R,t_T and t_B(tvalues).
   If t<tmin or t<tmax? ignore it and go to the next edge
   Otherwise classify the tvalue as entering or exiting value (using inner product to classify)
   If t is entering value set tmin=t if t is exiting value set tmax=t.</t</t

3. If tmin< tmax? then draw a line from (x1 + dx*tmin, y1 + dy*tmin) to (x1 + dx*tmax?, y1 + dy*tmax?)

4. If the line crosses over the window, you will see (x1 + dx*tmin, y1 + dy*tmin) and (x1 + dx*tmax? , y1 + dy*tmax?) are intersection between line and edge.

UNIT-III

Geometrical Transformation

Computer Graphics provide the facility of viewing object from different angles. The architect can study building from different angles i.e.

1. Front Evaluation
2. Side elevation
3. Top plan

A Cartographer can change the size of charts and topographical maps. So if graphics images are coded as numbers, the numbers can be stored in memory. These numbers are modified by mathematical operations called as Transformation.

The purpose of using computers for drawing is to provide facility to user to view the object from different angles, enlarging or reducing the scale or shape of object called as Transformation.

Two essential aspects of transformation are given below:

1. Each transformation is a single entity. It can be denoted by a unique name or symbol.
2. It is possible to combine two transformations, after connecting a single transformation is obtained, e.g., A is a transformation for translation. The B transformation performs scaling. The combination of two is C=AB. So C is obtained by concatenation property.

There are two complementary points of view for describing object transformation.

1. Geometric Transformation: The object itself is transformed relative to the coordinate system or background. The mathematical statement of this viewpoint is defined by geometric transformations applied to each point of the object.
2. Coordinate Transformation: The object is held stationary while the coordinate system is transformed relative to the object. This effect is attained through the application of coordinate transformations.

An example that helps to distinguish these two viewpoints:

The movement of an automobile against a scenic background we can simulate this by

- Moving the automobile while keeping the background fixed-(Geometric Transformation)
- We can keep the car fixed while moving the background scenery- (Coordinate Transformation)

Types of Transformations:

1. Translation
2. Scaling
3. Rotating
4. Reflection
5. Shearing

2D Transformation

Transformation means changing some graphics into something else by applying rules. We can have various types of transformations such as translation, scaling up or down, rotation, shearing, etc. When a transformation takes place on a 2D plane, it is called 2D transformation.

Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation.

Homogenous Coordinates

To perform a sequence of transformation such as translation followed by rotation and scaling, we need to follow a sequential process −

- Translate the coordinates,
- Rotate the translated coordinates, and then
- Scale the rotated coordinates to complete the composite transformation.

To shorten this process, we have to use 3×3 transformation matrix instead of 2×2 transformation matrix. To convert a 2×2 matrix to 3×3 matrix, we have to add an extra dummy coordinate W.

In this way, we can represent the point by 3 numbers instead of 2 numbers, which is called Homogenous Coordinate system. In this system, we can represent all the

transformation equations in matrix multiplication. Any Cartesian point P*X*,*Y* can be converted to homogenous coordinates by P' (X_h, Y_h, h).

Translation

A translation moves an object to a different position on the screen. You can translate a point in 2D by adding translation coordinate (t_x, t_y) to the original coordinate *X*,*Y* to get the new coordinate *X′*,*Y′*.

From the above figure, you can write that −

$$\mathbf{X' = X + t_x}$$

$$\mathbf{Y' = Y + t_y}$$

The pair (t_x, t_y) is called the translation vector or shift vector. The above equations can also be represented using the column vectors.

$$P=[X][Y]\ p' = [X'][Y']T = [tx][ty]$$

We can write it as −

$$\mathbf{P' = P + T}$$

Rotation

In rotation, we rotate the object at particular angle θ *theta* from its origin. From the following figure, we can see that the point P*X*,*Y* is located at angle φ from the horizontal X coordinate with distance r from the origin.
Let us suppose you want to rotate it at the angle θ. After rotating it to a new location, you will get a new point P' *X′*,*Y′*.

Using standard trigonometric the original coordinate of point PX,Y can be represented as −

$$X=r\cos\phi \quad .....(1)$$

$$Y=r\sin\phi \quad .....(2)$$

Same way we can represent the point P' X′,Y′ as −

$$x'=r\cos(\phi+\theta)=r\cos\phi\cos\theta-r\sin\phi\sin\theta \quad .....(3)$$

$$y'=r\sin(\phi+\theta)=r\cos\phi\sin\theta+r\sin\phi\cos\theta \quad .....(4)$$

Substituting equation 1 & 2 in 3 & 4 respectively, we will get

$$x'=x\cos\theta-y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

Representing the above equation in matrix form,

$[X'Y'] = [XY][\cos\theta \sin\theta - \sin\theta \cos\theta]$ OR

P' = P . R

Where R is the rotation matrix

$R = [\cos\theta \sin\theta - \sin\theta \cos\theta]$

The rotation angle can be positive and negative.

For positive rotation angle, we can use the above rotation matrix. However, for negative angle rotation, the matrix will change as shown below −

$R = [\cos(-\theta)\sin(-\theta) - \sin(-\theta)\cos(-\theta)]$
$= [\cos\theta - \sin\theta \sin\theta \cos\theta]$ ($\because \cos(-\theta) = \cos\theta \text{ and } \sin(-\theta) = -\sin\theta$)

Scaling

To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result.

Let us assume that the original coordinates are X, Y, the scaling factors are (S_X, S_Y), and the produced coordinates are X', Y'. This can be mathematically represented as shown below −

$$X' = X \cdot S_X \text{ and } Y' = Y \cdot S_Y$$

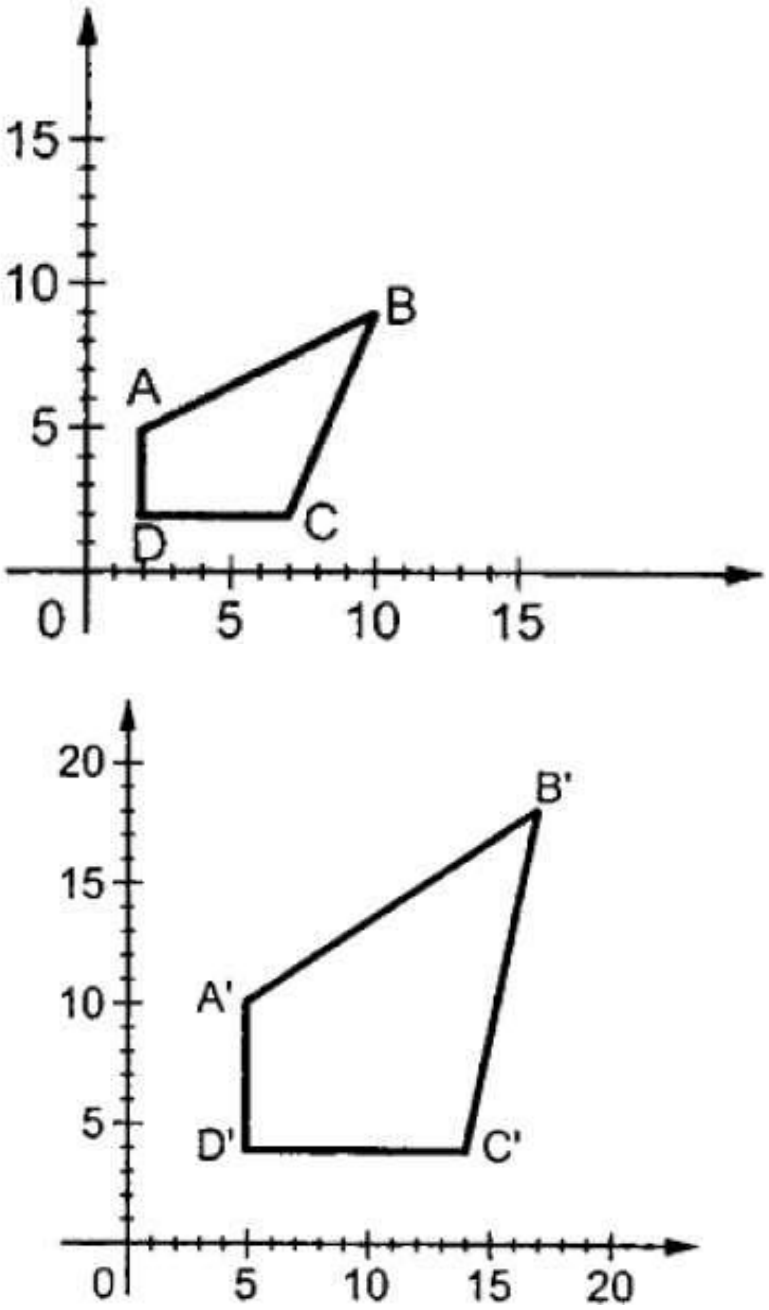
The scaling factor S_X, S_Y scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below −

$(X'Y') = (XY)[Sx\,0\,0\,Sy]$

OR

$$P' = P \cdot S$$

Where S is the scaling matrix. The scaling process is shown in the following figure.

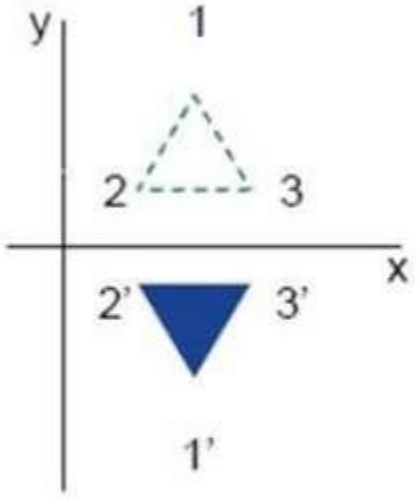


If we provide values less than 1 to the scaling factor S, then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object.

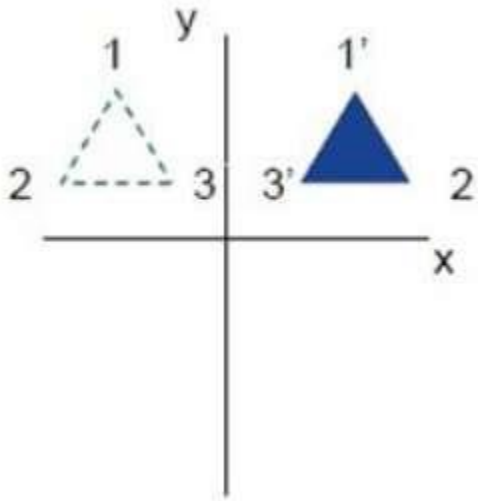
Reflection

Reflection is the mirror image of original object. In other words, we can say that it is a rotation operation with 180°. In reflection transformation, the size of the object does not change.

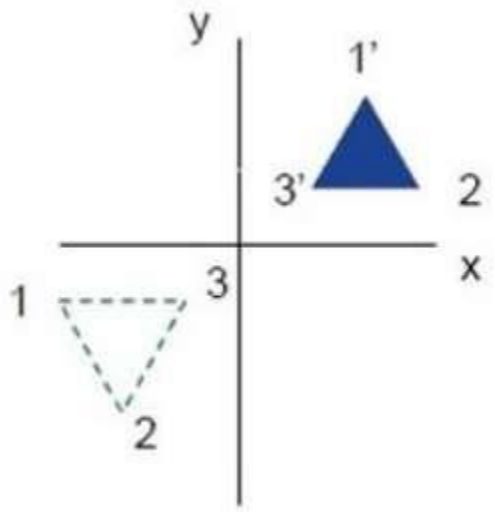
The following figures show reflections with respect to X and Y axes, and about the origin respectively.



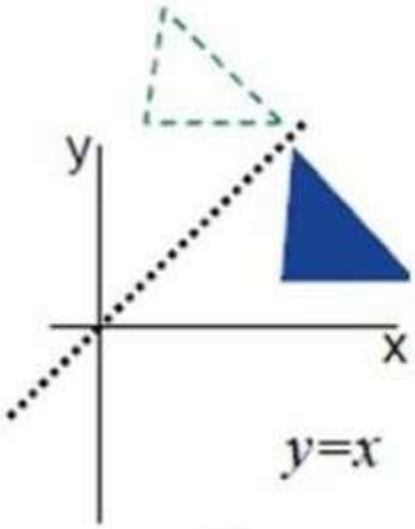
(a)



(b)



(c)



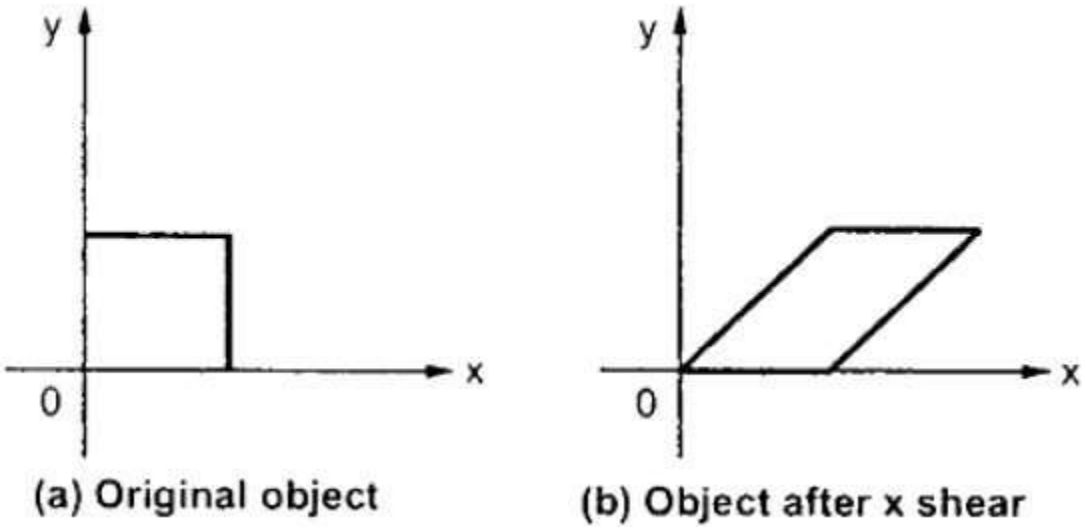
(d)

Shear

A transformation that slants the shape of an object is called the shear transformation. There are two shear transformations X-Shear and Y-Shear. One shifts X coordinates values and other shifts Y coordinate values. However; in both the cases only one coordinate changes its coordinates and other preserves its values. Shearing is also termed as Skewing.

X-Shear

The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.



(a) Original object (b) Object after x shear

The transformation matrix for X-Shear can be represented as −

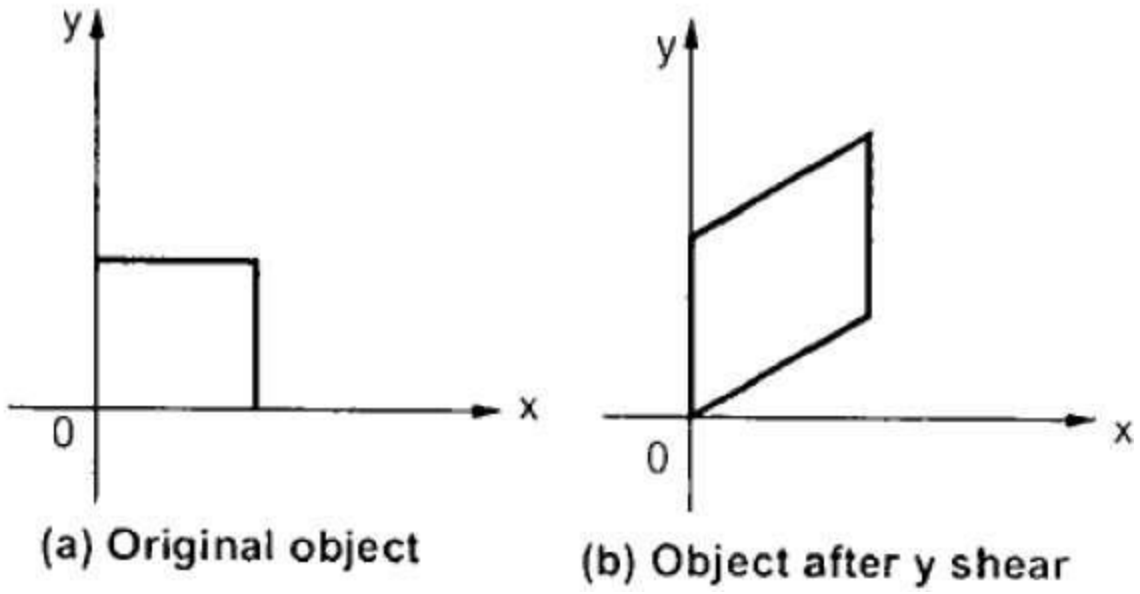
Xsh=[1shx0010001]

$$Y' = Y + Sh_y \cdot X$$

$$X' = X$$

Y-Shear

The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.



The Y-Shear can be represented in matrix from as −

Ysh[100*shy*10001]

$$X' = X + Sh_x \cdot Y$$

$$Y' = Y$$

Composite Transformation

If a transformation of the plane T1 is followed by a second plane transformation T2, then the result itself may be represented by a single transformation T which is the composition of T1 and T2 taken in that order. This is written as T = T1·T2.

Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.

A combined matrix −

[T][X] = [X] [T1] [T2] [T3] [T4] …. [Tn]

Where [Ti] is any combination of

- Translation
- Scaling
- Shearing
- Rotation
- Reflection

The change in the order of transformation would lead to different results, as in general matrix multiplication is not cumulative, that is [A] . [B] ≠ [B] . [A] and the order of multiplication. The basic purpose of composing transformations is to gain efficiency by applying a single composed transformation to a point, rather than applying a series of transformation, one after another.

For example, to rotate an object about an arbitrary point (X_p, Y_p), we have to carry out three steps −

- Translate point (X_p, Y_p) to the origin.
- Rotate it about the origin.
- Finally, translate the center of rotation back where it belonged.

Homogeneous Coordinates and Matrix Representation of 2D Transformations

Homogeneous Coordinates

The rotation of a point, straight line or an entire image on the screen, about a point other than origin, is achieved by first moving the image until the point of rotation occupies the origin, then performing rotation, then finally moving the image to its original position.

The moving of an image from one place to another in a straight line is called a translation. A translation may be done by adding or subtracting to each point, the amount, by which picture is required to be shifted.

Translation of point by the change of coordinate cannot be combined with other transformation by using simple matrix application. Such a combination is essential if we wish to rotate an image about a point other than origin by translation, rotation again translation.

To combine these three transformations into a single transformation, homogeneous coordinates are used. In homogeneous coordinate system, two-dimensional coordinate positions (x, y) are represented by triple-coordinates.

Homogeneous coordinates are generally used in design and construction applications. Here we perform translations, rotations, scaling to fit the picture into proper position.

Example of representing coordinates into a homogeneous coordinate system:

For two-dimensional geometric transformation, we can choose homogeneous parameter h to any non-zero value. For our convenience take it as one. Each two-dimensional position is then represented with homogeneous coordinates (x, y, 1).

Following are matrix for two-dimensional transformation in homogeneous coordinate:

1. Translation $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

2. Scaling $\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. Rotation (clockwise) $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

4. Rotation (anti-clock) $\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5. Reflection against X axis $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

6. Reflection against Y axis $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

7. Reflection against origin $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

8. Reflection against line Y=X $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

9. Reflection against Y= –X $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

10. Shearing in X direction $\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

11. Shearing in Y direction $\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

12. Shearing in both x and y direction $\begin{bmatrix} 1 & Sh_y & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Matrix Representation of 2D Transformation

1. Scaling $\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$

2. Rotation (clockwise) $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

3. Rotation (anti-clock) $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

4. Translation $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ t_x & t_y \end{bmatrix}$

5. Reflection $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

(about x axis)

6. Reflection $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

(about y axis)

7. Reflection $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

(about origin)

8. Reflection about Y=X $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

9. Reflection about Y= –X $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

10. Shearing in X direction $\begin{bmatrix} 1 & 0 \\ Sh_x & 1 \end{bmatrix}$

11. Shearing in Y direction $\begin{bmatrix} 1 & Sh_y \\ 0 & 1 \end{bmatrix}$

12. Shearing in both x and y direction $\begin{bmatrix} 1 & Sh_y \\ Sh_x & 1 \end{bmatrix}$

Program to implement 2-D Transformations:

```
#include<iostream.h>
#include<conio.h>
```

```
#include<math.h>
#include<stdlib.h>
#include<conio.h>
class trans
{
float x[20],y[20],xm,ym,ref[2][2],shx,shy;
int i,j,k,n;
float sx,sy,tx,ty,ang;
int gd,gm;
float xtmp [20],ytmp[20];
public:
void takeinput();
void menu();
void graphmode();
void mapgraph();
void plotint();
void translate();
void scale();
void rotate();
void reflect();
void shear();
void plotfinal();
};
int ch;
void trans::takeinput()
{
cout<<"ENTER THE NO OF VERTICES\n";
cin>>n;
for (i=0;i<n;i++)
{
cout<<"ENTER THE "<<i+1<<"COORDINATES \n";
cin>>x[i]>>y[i];
}
clrscr();
}
void trans::menu()
{
int kk;
cout<<"\n1:TRANSLATION";
cout<<"\n2:SCALING";
```

```
cout<<"\n3:ROTATION";
cout<<"\n4:REFLECTION";
cout<<"\n5:SHEARING";
cout<<"\n6:EXIT";
cin>>ch;
switch (ch)
{
case1:
cout<<"\n ENTER TX AND TY";
cin>>tx>>ty;
break;
case2:
cout<<"\n ENTER SX AND SY";
cin>>sx>>sy;
break;
case3:
cout<<"\n ENTER ANGLE OF ROTATION";
cin>>ang;
break;
case4:
cout<<"\n REFLECTION MENU";
cout<<"\n 1:X-PLANE";
cout<<"\n 2: Y-PLANE";
cout<<"\n 3: ORIGIN";
cout<<"\n 4: Y=X PLANE";
cout<<"\n 5: Y=-X PLANE";
cout<<"\n ENTER YOUR CHOICE";
cin>>kk;
switch (kk)
{
case1:
ref [0][0] =1;
ref [0][1]=0;
ref [1][0]=0;
ref [1][1]=1;
break;
case2:
ref [0][0]= -1;
ref [0][1]=0;
ref [1][0]=0;
```

```
ref [1][1]=1;
break;
case3:
ref [0][0]=-1;
ref [0][1]=0;
ref [1][0]=0;
ref [1][1]=1;
break;
case4:
ref [0][0]=0;
ref [0][1]=1;
ref [1][0] =1;
ref [1][1]=0;
break;
case5:
ref [0][0]=0;
ref [0][1]=1;
ref [1][0]=1;
ref [1][1]=0;
break;
case5:
cout<< "\n SHEARING MENU";
cout<<"\n 1:X-DIR\n 2: Y-DIR \n 3: X-Y DIR\n ENTER YOUR              CHOICE";
cin>>kk;
switch (kk)
{
case1:
cout<<"\n ENTER SHX";
cin>> shx;
ref[0][0] =1;
ref [0][1]=0;
ref [1][0]=shx;
ref [1][1]=1;
break;
case2:
cout<< "\n ENTER SHY";
cin>>shy;
ref [0][0]=1;
ref [0][1]=shy;
ref [1][0]=0;
```

```
ref [1][1] =1;
break;
case3:
cout<<"\n ENTER SHX";
cin >> shx;
cout<<"\n ENTER SHY";
cin>> shy;
ref [0][0] =1;
ref [0][1] =shy;
ref [1][0] =shx;
ref [1][1] =1;
break;
}
break;
}
}
void trans::graphmode()
{
gd=DETECT;
initgraph (&gd, &gm, "");
}
void trans::mapgraph()
{
xm=getmaxx ()/2;
ym=getmaxy ()/2;
line (xm,0,xm,2*ym);
line (0,ym,2 * xm,ym);
}
void trans::plotint()
{
for(i=0;i<n-1;i++)
{
circle (x[i] +xm,-y[i]+ym,2)
circle x [n-1]+xm,(-y[n-1]+ym),2;
line (x[i]+xm,(-y[i]+ym),x[i+1]+xm,(-y[i+1]+ym));
}
line (x[n-1]+xm,(-y[n-1]+ym,)x[0]+xm,(-y[0]+ym));
}
void trans::translate()
{
```

```
for(i=0;i<n;i++)
{
xtmp[i]=x[i]+tx;
ytmp[i]=y[i]+ty;
}
}
void trans::plotfinal()
{
for (i=0;i<n-1;i++)
{
circle (xtmp[i]+xm, (-ytmp[i]+ym,2);
circle (xtmp[n-1]+xm,(-ytmp[n-1]+ym),2);
line (xtmp[i]+xm,(-ytmp[i]+ym),xtmp[i+1]+xm,(-ytmp[i+1]+ym));
}
line (xtmp[n-1]+xm,(-ytmp[n-1]+ym),xtmp[0]+xm,(-ytmp[0]+ym));
}
void trans::scale()
{
float s [2][2],mxy[7][2],rxy[7][2];
s [0][0]=sx;
s [0][1]=0;
s [1][0]=0;
s [1][1]=sy;
tx=-x[0];
ty=-y[0];
translate ();
k=0;
for(i=0;i<n;i++)
{
j=0;
mxy[i][j]=xtmp[k];
mxy[i][j+1]=ytmp[k];
k++;
}
for (i=0;i<n;i++)
{
for(j=0;j<2;j++)
{
rxy[i][j]=0;
for(k=0;k<2;k++)
```

```
{
rxy[i][j]+=mxy[i][k]*s[k][j];
}
}
}
j=0;
k=0;
for(i=0;i<n;i++)
{
j=0;
x[k]=rxy[i][j];
y[k]=rxy[i][j+1];
k++;
}
tx=-tx;
ty=-ty;
translate();
}
void trans::rotate()
{
float r[2][2],mxy[7][2],rxy[7][2],tmp;
tmp=22/7;
tmp=(tmp*ang)/180;
r[0][0]=cos(tmp);
r[0][1]=sin(tmp);
r[1][0]=cos(tmp);
r[1][1]=sy;
tx=-x[0];
ty=-y[0];
translate ();
k=0;
for (i=0;i<n;i++)
{
j=0;
mxy[i][j]=xtmp[k];
mxy[i][j+1]=ytmp[k];
k++;
}
for (i=0;i<n;i++)
{
```

```
for (j=0;j<2;j++)
{
rxy[i][j]=0;
for (k=0;k<2;k++)
{
rxy[i][j]+=mxy[i][k]*r[k][j];
}
}
}
j=0;
k=0;
for(i=0;i<n;i++)
{
j=0;
x[k]=rxy[i][j];
y[k]=rxy[i][j+1];
k++;
}
tx=-tx;
ty=-ty;
translate();
}
void trans::reflect()
{
float mxy[7][2],rxy[7][2],tmp;
tx=0;
ty=0;
translate();
k=0;
for(i=0;i<n;i++)
{
j=0;
mxy[i][j]=xtmp[k];
mxy[i][j+1]=ytmp[k];
k++;
}
for(i=0;i<n;i++)
{
for(j=0;j<2;j++)
{
```

```
rxy[i][j]=0;
for(k=0;k<2;k++)
{
rxy[i][j]|+=mxy[i][k]*r[k][j];
}
}
}
j=0;
k=0;
for(i=0;i<n;i++)
{
j=0;
x[k]=rxy[i][j];
y[k]=rxy[i][j+1];
k++;
}
tx=-tx;
ty=-ty;
translate();
}
void trans::shear()
{
float mxy[7][2],rxy[7][2],tmp;
tx=0;
ty=0;
translate ();
k=0;
for(i=0;i<n;i++)
{
j=0;
mxy[i][j]=xtmp[k];
mxy[i][j+1]=ytmp[k];
k++;
}
for(i=0;i<n;i++)
{
for(j=0;j<2;j++)
{
rxy[i][j]=0;
for (k=0;k<2;k++)
```

```
{
rxy[i][j]|+=mxy[i][k]*r[k][j];
}
}
}
j=0;
k=0;
for(i=0;i<n;i++)
{
j=0;
x[k]=rxy[i][j];
y[k]=rxy[i][j+1];
k++;
}
tx=-tx;
ty=-ty;
translate ();
}
void main()
{
clrscr ();
trans t1;
t1.takeinput ();
t1.menu ();
t1.graphmode ();
t1.mapgraph ();
t1.plotint ();
switch (ch)
{
case1:
t1.translate ();
break;
case2:
t1.scale ();
break ();
case3:
t1.rotate ();
break;
case4:
t1.reflect ();
```

```
break;
case5:
t1.shear ();
break;
case6:
exit ();
}
getch ();
t1.plotfinal ();
getch ();
closegraph ();
}
```

Output:

Translate

```
1: TRANSLATION
2: SCALING
3: ROTATION
4: REFLECTION
5: SHEARING
6: EXIT

ENTER YOUR CHOICE 4

REFLECTION MENU

1: X-PLANE
2: Y-PLANE
3: ORIGIN
4: Y=X PLANE
5: Y=-X PLANE

ENTER YOUR CHOICE 4

1: TRANSLATION
2: SCALING
3: ROTATION
4: REFLECTION
```

5: SHEARING
6: EXIT

ENTER YOUR CHOICE 5

SHEARING MENU

1: X-DIR
2: Y-DIR

ENTER YOUR CHOICE 3

ENTER SHX 5
ENTER SHY 5
ENTER THE NO OF VERTICES
5
ENTER THE 1 COORDINATES
10 10
ENTER THE 2 COORDINATES
30 10
ENTER THE 3 COORDINATES
40 20
ENTER THE 4 COORDINATES
35 30
ENTER THE 5 COORDINATES
15 20
1: TRANSLATION
2: SCALING
3: ROTATION
4: REFLECTION
5: SHEARING
6: EXIT
ENTER YOUR CHOICE 1
ENTER TX AND TY 10 10

Composition of 2D Transformations

Prerequisite – Basic types of 2-D Transformation :

1. Translation
2. Scaling
3. Rotation
4. Reflection
5. Shearing of a 2-D object

Composite Transformation :

As the name suggests itself Composition, here we combine two or more transformations into one single transformation that is equivalent to the transformations that are performed one after one over a 2-D object.

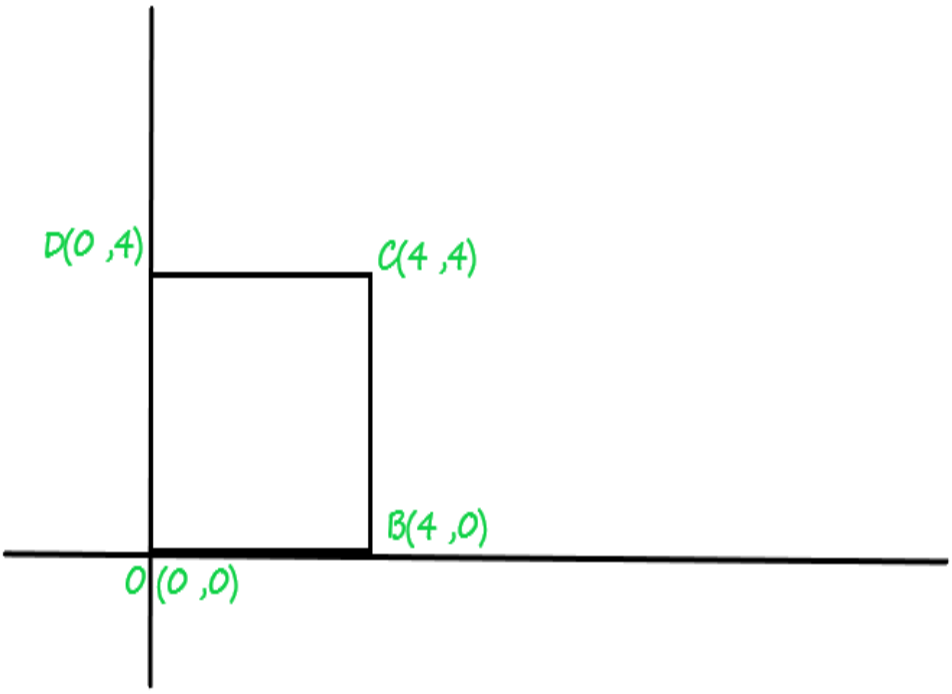
Example :

Consider we have a 2-D object on which we first apply transformation T_1 (2-D matrixcondition) and then we apply transformation T_2(2-D matrix condition) over the 2-D object and the object get transformed, the very equivalent effect over the 2-D object we can obtain by multiplying T_1 & T_2 (2-D matrix conditions) with each other and then applying the T_{12} (resultant of T_1 X T_2) with the coordinates of the 2-D image to get the transformed final image.

Problem :

Consider we have a square O(0, 0), B(4, 0), C(4, 4), D(0, 4) on which we first apply T1(scaling transformation) given scaling factor is Sx=Sy=0.5 and then we apply T2(rotatation transformation in clockwise direction) it by 90^*(angle), in last we perform T3(reflection transformation about origin).

Ans : The square O, A, C, D looks like :



Square_given(Fig.1)

Window-to-Viewport Transformations

Window:

1. A world-coordinate area selected for display is called a window.
2. In computer graphics, a window is a graphical control element.
3. It consists of a visual area containing some of the graphical user interface of the program it belongs to and is framed by a window decoration.
4. A window defines a rectangular area in world coordinates. You define a window with a GWINDOW statement. You can define the window to be larger than, the same size as, or smaller than the actual range of data values, depending on whether you want to show all of the data or only part of the data.

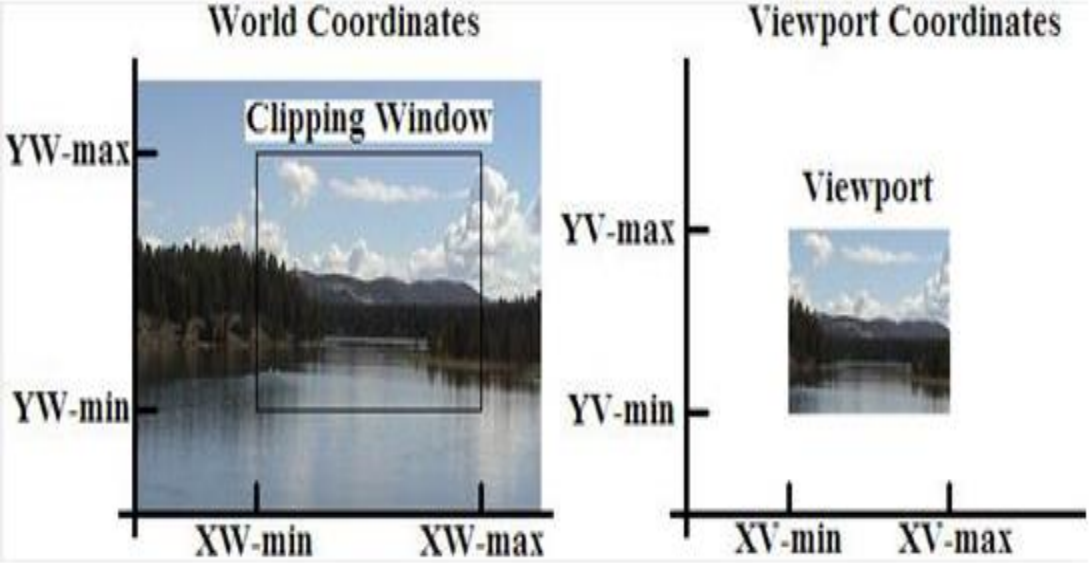
Viewport:

1. An area on a display device to which a window is mapped is called a viewport.
2. A viewport is a polygon viewing region in computer graphics. The viewport is an area expressed in rendering-device-specific coordinates, e.g. pixels for screen coordinates, in which the objects of interest are going to be rendered.

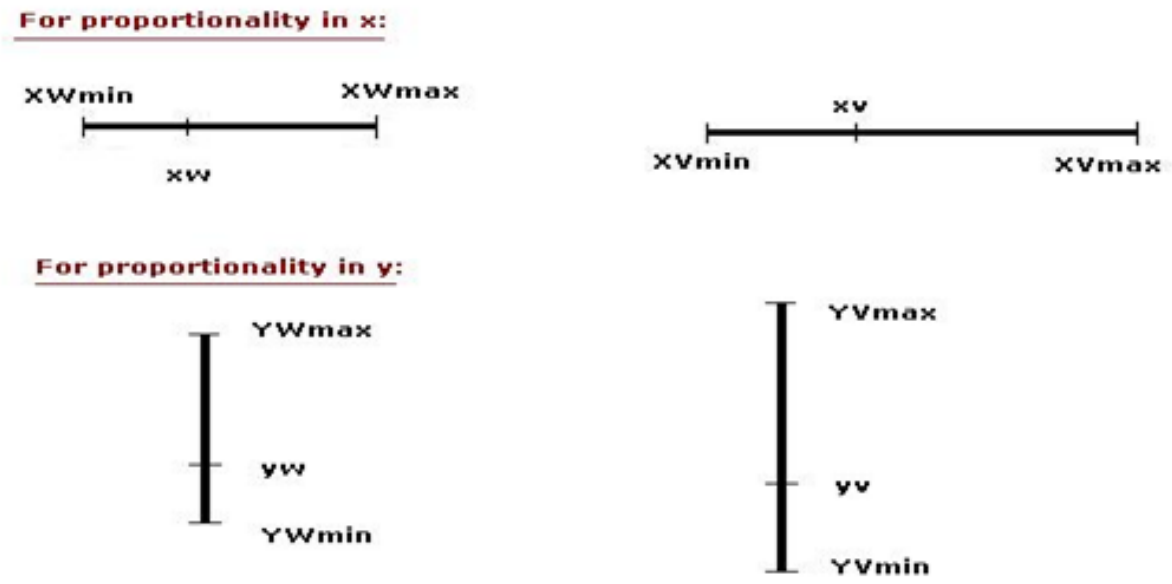
3. A viewport defines in normalized coordinates a rectangular area on the display device where the image of the data appears. You define a viewport with the GPORT command. You can have your graph take up the entire display device or show it in only a portion, say the upper-right part.

Window to viewport transformation:

1. Window-to-Viewport transformation is the process of transforming a two-dimensional, world-coordinate scene to device coordinates.
2. In particular, objects inside the world or clipping window are mapped to the viewport. The viewport is displayed in the interface window on the screen.
3. In other words, the clipping window is used to select the part of the scene that is to be displayed. The viewport then positions the scene on the output device.
4. **Example:**
5.



1. This transformation involves developing formulas that start with a point in the world window, say (xw, yw).
2. The formula is used to produce a corresponding point in viewport coordinates, say (xv, yv). We would like for this mapping to be "proportional" in the sense that if xw is 30% of the way from the left edge of the world window, then xv is 30% of the way from the left edge of the viewport.
3. Similarly, if yw is 30% of the way from the bottom edge of the world window, then yv is 30% of the way from the bottom edge of the viewport. The picture below shows this proportionality.



1. Using this proportionality, the following ratios must be equal.

$$\frac{xv - xvmin}{xvmax - xvmin} = \frac{xw - xwmin}{xwmax - xwmin}$$

$$\frac{yv - yvmin}{yvmax - yvmin} = \frac{yw - ywmin}{ywmax - ywmin}$$

1. By solving these equations for the unknown viewport position (xv, yv), the following becomes true:

$$xv = Sxxw + tx$$

$$yv = Syyw + ty$$

1. The scale factors (Sx, Sy) would be:

$$Sx = \frac{xvmax - xvmin}{xwmax - xwmin}$$

$$Sy = \frac{yvmax - yvmin}{ywmax - ywmin}$$

1. And the translation factors (Tx, Ty) would be:

$$tx = \frac{xwmax\,xvmin - xwmin\,xvmax}{xwmax - xwmin}$$

$$ty = \frac{ywmax\,yvmin - ywmin\,yvmax}{ywmax - ywmin}$$

1. The position of the viewport can be changed allowing objects to be viewed at different positions on the Interface Window.
2. Multiple viewports can also be used to display different sections of a scene at different screen positions. Also, by changing the dimensions of the viewport, the size and proportions of the objects being displayed can be manipulated.
3. Thus, a zooming affect can be achieved by successively mapping different dimensioned clipping windows on a fixed sized viewport.
4. If the aspect ratio of the world window and the viewport are different, then the image may look distorted.

Introduction to 3D Transformations Matrix

The geometric transformations play a vital role in generating images of three Dimensional objects with the help of these transformations. The location of objects relative to others can be easily expressed. Sometimes viewpoint changes rapidly, or sometimes objects move in relation to each other. For this number of transformation can be carried out repeatedly.

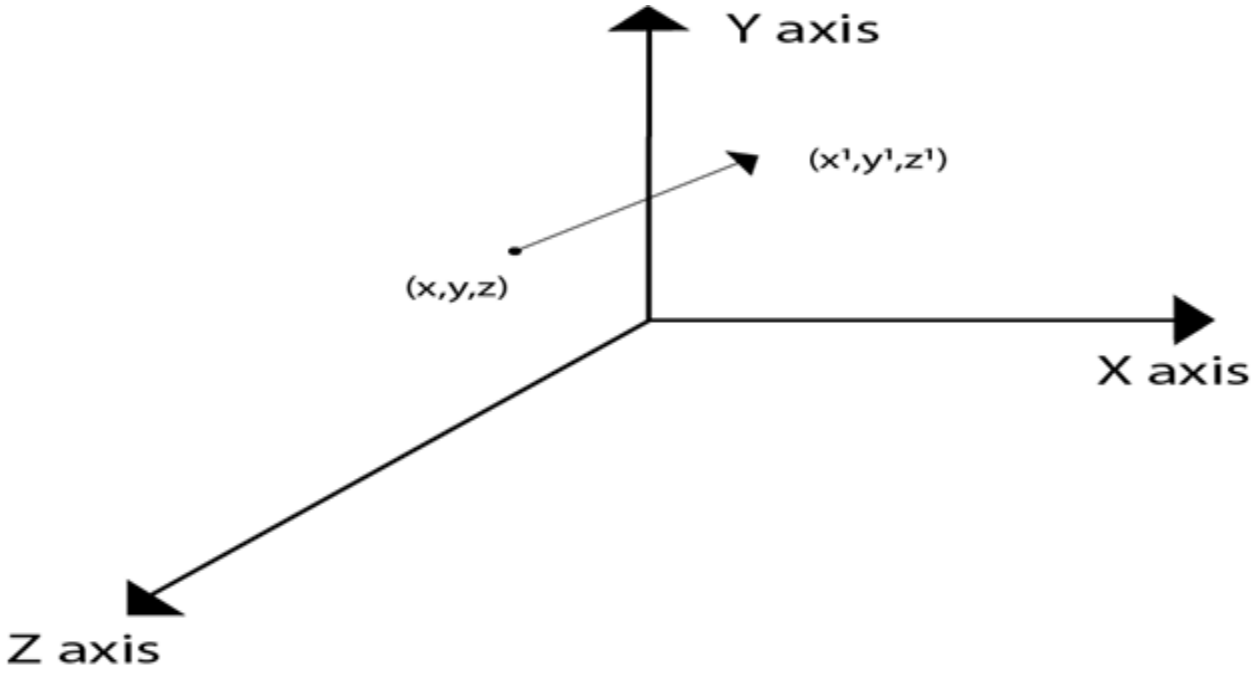
Translation

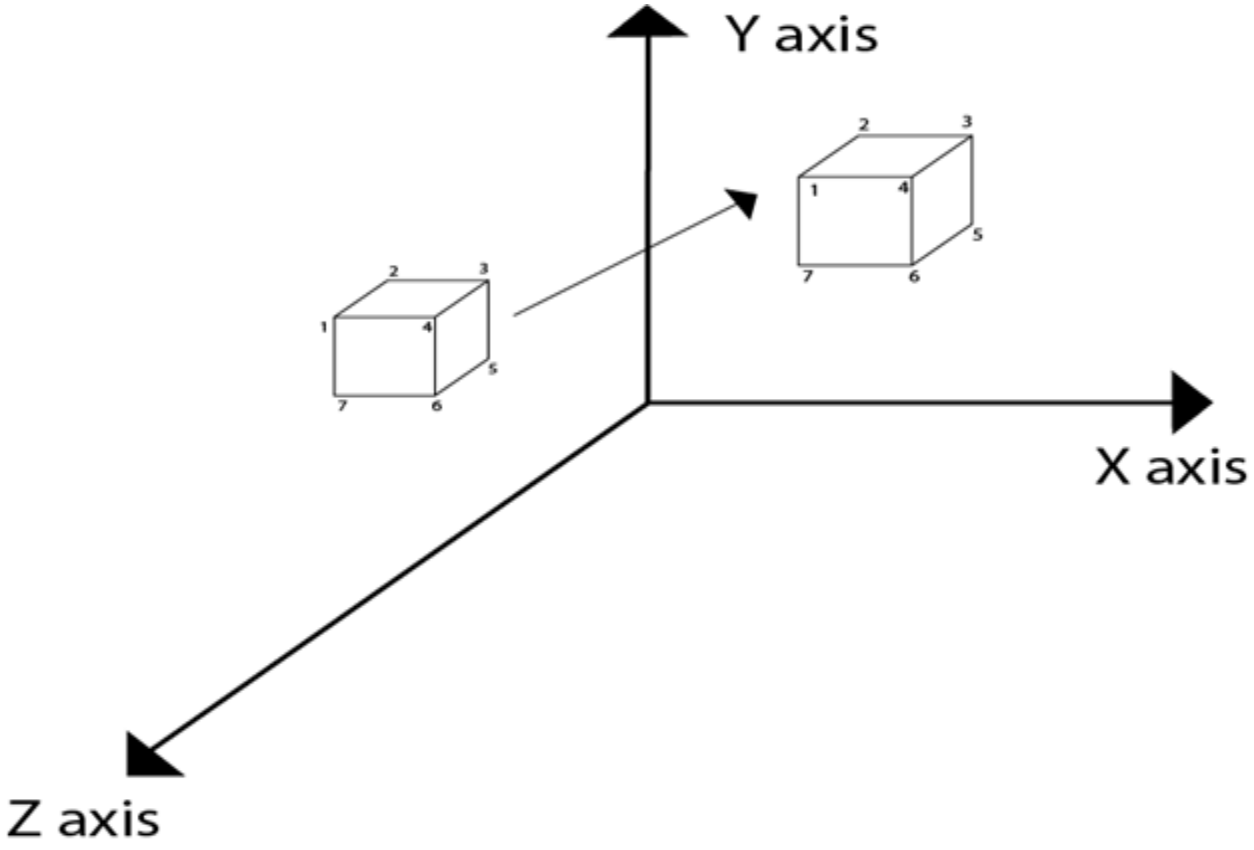
It is the movement of an object from one position to another position. Translation is done using translation vectors. There are three vectors in 3D instead of two. These vectors are in x, y, and z directions. Translation in the x-direction is represented using T_x. The translation is y-direction is represented using T_y. The translation in the z-direction is represented using T_z.

If P is a point having co-ordinates in three directions (x, y, z) is translated, then after translation its coordinates will be (x^1 y^1 z^1) after translation. T_x T_y T_z are translation vectors in x, y, and z directions respectively.

$$x^1 = x + T_x$$
$$y^1 = y + T_y$$
$$z^1 = z + T_z$$

Three-dimensional transformations are performed by transforming each vertex of the object. If an object has five corners, then the translation will be accomplished by translating all five points to new locations. Following figure 1 shows the translation of point figure 2 shows the translation of the cube.


Y axis
(x¹,y¹,z¹)
(x,y,z)
X axis
Z axis


Y axis
2
3
1
4
5
7
6
X axis
Z axis

Matrix for translation

$$\begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{Bmatrix} \text{ or } \begin{Bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

Matrix representation of point translation

Point shown in fig is (x, y, z). It become (x^1,y^1,z^1) after translation. T_x T_y T_z are translation vector.

$$\begin{pmatrix} x^1 \\ y^1 \\ z^1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Example: A point has coordinates in the x, y, z direction i.e., (5, 6, 7). The translation is done in the x-direction by 3 coordinate and y direction. Three coordinates and in the z-direction by two coordinates. Shift the object. Find coordinates of the new position.

Solution: Co-ordinate of the point are (5, 6, 7)
Translation vector in x direction = 3
Translation vector in y direction = 3
Translation vector in z direction = 2

Translation matrix is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

Multiply co-ordinates of point with translation matrix

$$(x^1 y^1 z^1) = (5,6,7,1) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3 & 3 & 2 & 1 \end{pmatrix}$$

= [5+0+0+30+6+0+30+0+7+20+0+0+1] = [8991]

x becomes x^1=8
y becomes y^1=9
z becomes z^1=9

UNIT-IV

Representing Curves & Surfaces

In computer graphics, we often need to draw different types of objects onto the screen. Objects are not flat all the time and we need to draw curves many times to draw an object.

Types of Curves

A curve is an infinitely large set of points. Each point has two neighbors except endpoints. Curves can be broadly classified into three categories − explicit, implicit, and parametric curves.

Implicit Curves

Implicit curve representations define the set of points on a curve by employing a procedure that can test to see if a point in on the curve. Usually, an implicit curve is defined by an implicit function of the form −

fx,yx,y = 0
It can represent multivalued
curves multipleyvaluesforanxvaluemultipleyvaluesforanxvalue. A common example is the circle, whose implicit representation is

$x^2 + y^2 - R^2 = 0$

Explicit Curves
A mathematical function y = fxx can be plotted as a curve. Such a function is the explicit representation of the curve. The explicit representation is not general, since it cannot represent vertical lines and is also single-valued. For each value of x, only a single value of y is normally computed by the function.

Parametric Curves

Curves having parametric form are called parametric curves. The explicit and implicit curve representations can be used only when the function is known. In practice the parametric curves are used. A two-dimensional parametric curve has the following form −

Ptt = ftt, gtt or Ptt = xtt, ytt

The functions f and g become the x,yx,y coordinates of any point on the curve, and the points are obtained when the parameter t is varied over a certain interval [a, b], normally [0, 1].

Bezier Curves

Bezier curve is discovered by the French engineer Pierre Bézier. These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve. The Bezier curve can be represented mathematically as −

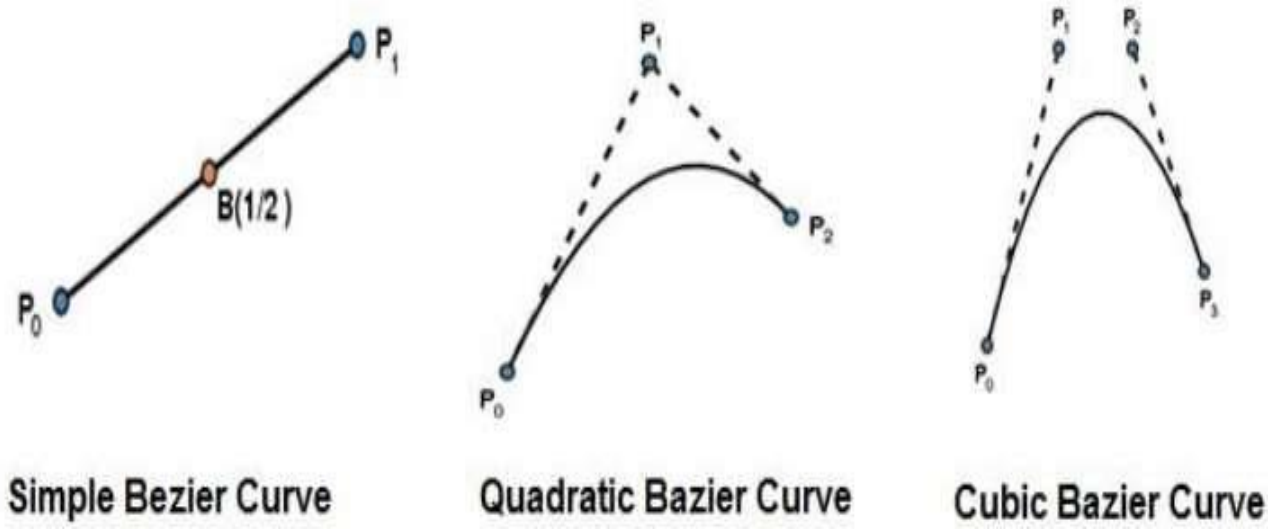
∑k=0nPiBni(t)∑k=0nPiBin(t)

Where pipi is the set of points and Bni(t)Bin(t) represents the Bernstein polynomials which are given by −

Bni(t)=(ni)(1−t)n−itiBin(t)=(ni)(1−t)n−iti

Where **n** is the polynomial degree, **i** is the index, and **t** is the variable.

The simplest Bézier curve is the straight line from the point P0P0 to P1P1. A quadratic Bezier curve is determined by three control points. A cubic Bezier curve is determined by four control points.



Properties of Bezier Curves

Bezier curves have the following properties −

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.

- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

B-Spline Curves

The Bezier-curve produced by the Bernstein basis function has limited flexibility.

- First, the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
- The second limiting characteristic is that the value of the blending function is nonzero for all parameter values over the entire curve.

The B-spline basis contains the Bernstein basis as the special case. The B-spline basis is non-global.

A B-spline curve is defined as a linear combination of control points Pi and B-spline basis function Ni,Ni, k tt given by
$C(t)=\sum ni=0PiNi,k(t),C(t)=\sum i=0nPiNi,k(t)$, $n\geq k-1,n\geq k-1$, $t\epsilon[tk-1,tn+1]t\epsilon[tk-1,tn+1]$

Where,

- {pipi: i=0, 1, 2….n} are the control points
- k is the order of the polynomial segments of the B-spline curve. Order k means that the curve is made up of piecewise polynomial segments of degree k - 1,

- the Ni,k(t)Ni,k(t) are the “normalized B-spline blending functions”. They are described by the order k and by a non-decreasing sequence of real numbers normally called the “knot sequence”.

ti:i=0,...n+Kti:i=0,...n+K

The N_i, k functions are described as follows −

Ni,1(t)={1,0,ifuϵ[ti,ti+1)OtherwiseNi,1(t)={1,ifuϵ[ti,ti+1)0,Otherwise

and if k > 1,

Ni,k(t)=t−titi+k−1Ni,k−1(t)+ti+k−tti+k−ti+1Ni+1,k−1(t)Ni,k(t)=t−titi+k−1Ni,k−1(t)+ti+k−tti+k−ti+1Ni+1,k−1(t)

and

tϵ[tk−1,tn+1)tϵ[tk−1,tn+1)

Properties of B-spline Curve

B-spline curves have the following properties −

- The sum of the B-spline basis functions for any parameter value is 1.
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for k=1.
- The maximum order of the curve is equal to the number of vertices of defining polygon.
- The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.
- The curve exhibits the variation diminishing property.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon.

Polygon Surfaces

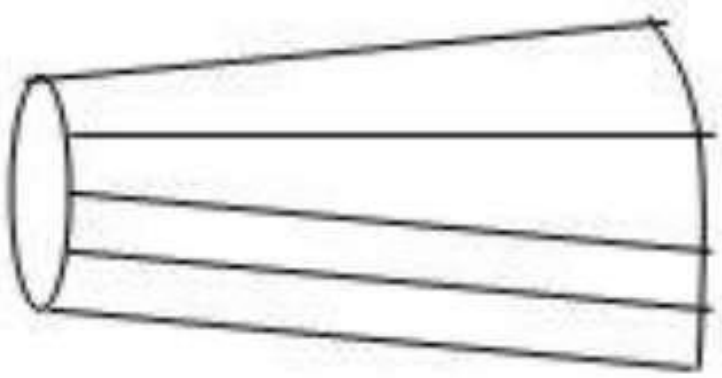
Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

- **Boundary Representations** B−repsB−reps − It describes a 3D object as a set of surfaces that separates the object interior from the environment.

- **Space–partitioning representations** − It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids usuallycubesusuallycubes.

The most commonly used boundary representation for a 3D graphics object is a set of surface polygons that enclose the object interior. Many graphics system use this method. Set of polygons are stored for object description. This simplifies and speeds up the surface rendering and display of object since all surfaces can be described with linear equations.

The polygon surfaces are common in design and solid-modeling applications, since their wireframe display can be done quickly to give general indication of surface structure. Then realistic scenes are produced by interpolating shading patterns across polygon surface to illuminate.



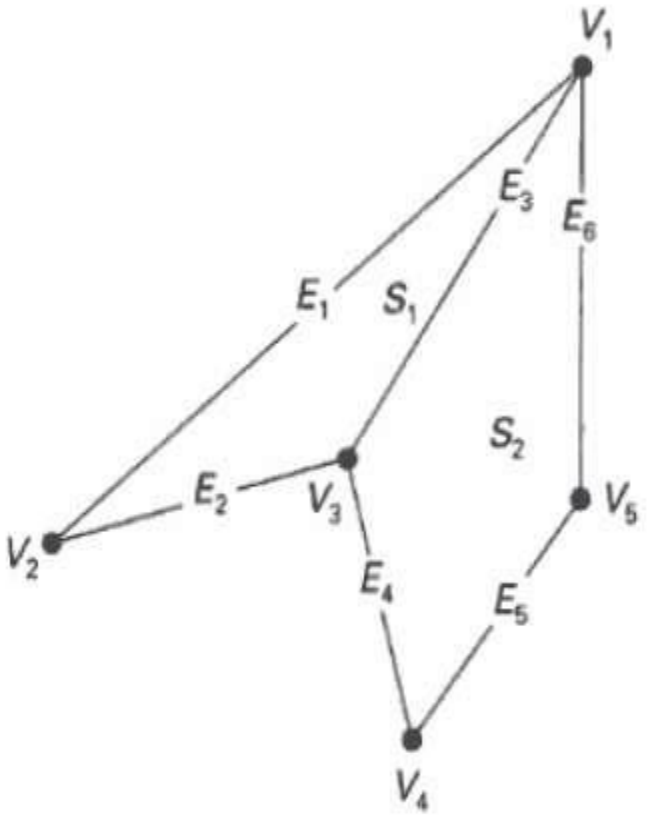
A 3D object represented by polygons

Polygon Tables

In this method, the surface is specified by the set of vertex coordinates and associated attributes. As shown in the following figure, there are five vertices, from v_1 to v_5.

- Each vertex stores x, y, and z coordinate information which is represented in the table as v_1: x_1, y_1, z_1.

- The Edge table is used to store the edge information of polygon. In the following figure, edge E_1 lies between vertex v_1 and v_2 which is represented in the table as E_1: v_1, v_2.
- Polygon surface table stores the number of surfaces present in the polygon. From the following figure, surface S_1 is covered by edges E_1, E_2 and E_3 which can be represented in the polygon surface table as S_1: E_1, E_2, and E_3.



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

Plane Equations

The equation for plane surface can be expressed as −

$$Ax + By + Cz + D = 0$$

Where x,y,zx,y,z is any point on the plane, and the coefficients A, B, C, and D are constants describing the spatial properties of the plane. We can obtain the values of A, B, C, and D by solving a set of three plane equations using the coordinate values for three non collinear points in the plane. Let us assume that three vertices of the plane are (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3).

Let us solve the following simultaneous equations for ratios A/D, B/D, and C/D. You get the values of A, B, C, and D.

$$A/DA/D\ x_1 + B/DB/D\ y_1 + C/DC/D\ z_1 = -1$$
$$A/DA/D\ x_2 + B/DB/D\ y_2 + C/DC/D\ z_2 = -1$$
$$A/DA/D\ x_3 + B/DB/D\ y_3 + C/DC/D\ z_3 = -1$$

To obtain the above equations in determinant form, apply Cramer's rule to the above equations.

A=[[[111y1y2y3z1z2z3]]]B=[[[x1x2x3111z1z2z3]]]C=[[[x1x2x3y1y2y3111]]]D=−[[[x1x2x3y1y2y3z1z2z3]]]A=[1y1z11y2z21y3z3]B=[x11z1x21z2x31z3]C=[x1y11x2y21x3y31]D=−[x1y1z1x2y2z2x3y3z3]

For any point x,y,zx,y,z with parameters A, B, C, and D, we can say that −

- $Ax + By + Cz + D \neq 0$ means the point is not on the plane.
- $Ax + By + Cz + D < 0$ means the point is inside the surface.
- $Ax + By + Cz + D > 0$ means the point is outside the surface.

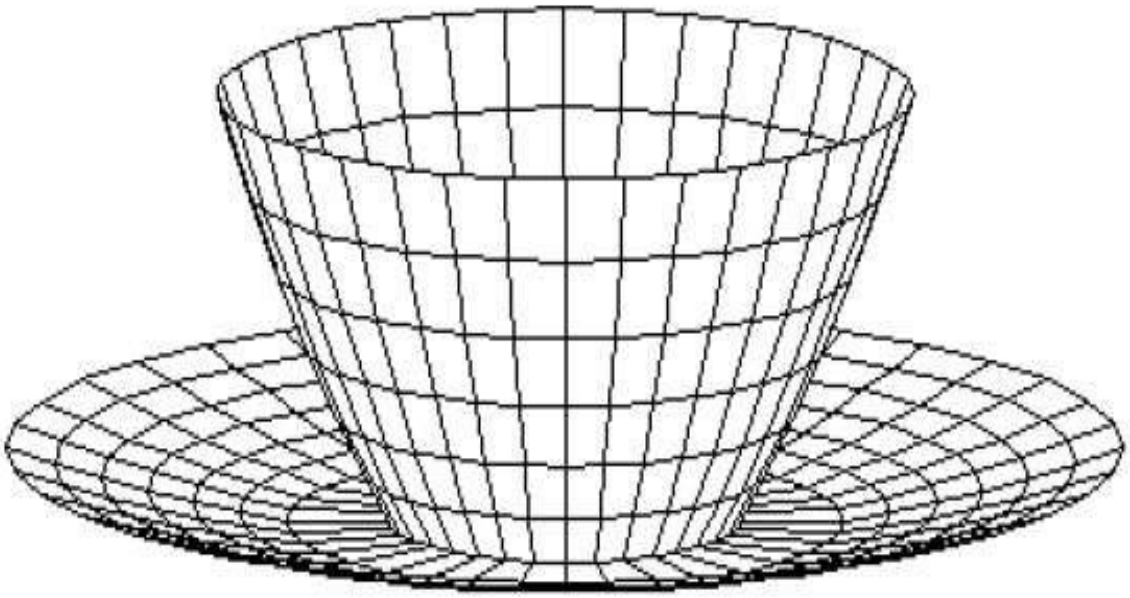
Polygon Meshes

3D surfaces and solids can be approximated by a set of polygonal and line elements. Such surfaces are called polygonal meshes. In polygon mesh, each edge is shared by at most two polygons. The set of polygons or faces, together form the "skin" of the object.

This method can be used to represent a broad class of solids/surfaces in graphics. A polygonal mesh can be rendered using hidden surface removal algorithms. The polygon mesh can be represented by three ways −

- Explicit representation
- Pointers to a vertex list

- Pointers to an edge list



Advantages

- It can be used to model almost any object.
- They are easy to represent as a collection of vertices.
- They are easy to transform.
- They are easy to draw on computer screen.

Disadvantages

- Curved surfaces can only be approximately described.
- It is difficult to simulate some type of objects like hair or liquid.

Polygon meshes parametric

In computer graphics a polygon mesh is the collection of vertices, edges, and faces that make up a 3D object. A polygon mesh defines the shape and contour of every 3D character & object, whether it be used for 3D animated film, advertising, or video games.

While the concept may seem daunting, the geometry behind a polygon mesh is easy to understand.

Large forms are built from smaller, interconnected planes—usually triangles or rectangles—that fit together like a 3D jigsaw puzzle.

Each vertex in the polygon mesh stores x, y, and z coordinate information.

Then each face of that polygon contains surface information which is used by the rendering engine to calculate lightning and shadows(among other things).

Polygon meshes can be used to model almost any object.

Note it's also possible to generate polygon meshes in real time making them both powerful and dynamic. As such, they are used extensively throughout the world of computer graphics.

Digging Deeper Into Meshes

The idea behind modeling a polygon mesh is to approximate the 3D surface of anything with lines and polygons.

Artists are then free to add as much detail as they choose by increasing the number of vertices in the mesh. The only limit is the capabilities of the computer.

In the 3D world are several programs that create polygon meshes.

The most popular programs are things like Blender, Maya, and 3ds Max. These all provide tools for modeling, texturing, rigging, and animating 3D meshes.

These can all be customized for use in video games or films depending on the project.

Although most 3D objects are solid, polygon meshes don't necessarily have to be.

A single plane can be used to create a "thin" mesh. For optimization, most meshes are rendered as polygonal quads—four connected vertices—that are split into triangles by the computer.

Each face of any polygon mesh has two sides: a front face and a back face.

The front face is used to calculate the surface angle of the mesh while the back face is meant to be hidden from the camera.

If something goes wrong and the camera sees a back face it will be rendered incorrectly. This might be the cause of some old glitch blocks you'd see in classic PlayStation or N64 games.

Meshes also contain data about their UV coordinates. These are used to display textures correctly on the surface of the mesh.

By UV-unwrapping the mesh(think of flattening or unfolding a box) artists can paint the surface with textures and color, then re-form the 3D shape after.

Polygonal meshes do have their limitations though. It is difficult to approximate curved surfaces with a series of lines. Organic shapes require a large number of vertices. Objects like hair and liquid are also very difficult to simulate using polygon meshes.

Looking Behind the Scenes

All our favorite video games and 3D cartoon characters are made from meshes.

To create life these meshes can be "deformed" to make them move, twist, or turn. In this way characters can be made to walk for fly across the screen.

In most studios an artist will make a mesh and pass it onto another artist for rigging and animation. Each artist specializes in a different part of the 3D animation pipeline but they all work with the same assets.

To create the mesh, most artists will start with a drawing that shows the object they want to make from multiple angles. Usually they will have at least the front and side views. By switching between these views while modeling, the artist can better create the 3d form with more accuracy.

It's usually best to start by modeling a low-poly version of the entire object. This means reducing the details to approximate the most general forms. This low-poly version of the model can be used to create various high-resolution versions later.

Having different versions of the model is super useful for video game development where processing power is scarce. To save memory, most studios create games where they only show the high-poly model when the camera is close. When the object is far away, the low-poly version is substituted.

Without a texture the polygonal mesh won't look like much.

Adding some color will keep it from looking like a boring statue but it's still just a basic mesh.

With textures, artists can change the appearance of their models to look like any variety of surfaces, including dirt or skin or blue jeans.

And by using UV coordinates textures can be applied to the surface of models. By carefully mapping the places of the mesh onto an image, artists can give each plane a unique texture as well.

This UV map can be brought into any image processing program and edited to whatever detail is needed for the project.

It's also possible to combine different textures in some programs. For example, by overlaying a color map with a specular map you can give the appearance of surface details.

We've only scratched the surface of what's possible using a polygon mesh. Most commercial products are built using CAD and other mesh creation software which makes everything much easier to learn if you're new to 3D work.

In many ways the modern CG world has been built using polygon meshes.

If you've ever watched anything animated in 3D then you've seen a lot of them.

Cubic Curves

A cubic curve is an Algebraic Curve of degree 3. An algebraic curve over a Field K is an equation $f(X,Y)=0$, where $f(X,Y)$ is a Polynomial in X and Y with Coefficients in K, and the degree of f is the Maximum degree of each of its terms (Monomials).

Newton showed that all cubics can be generated by the projection of the five divergent cubic parabolas. Newton's classification of cubic curves appeared in the chapter ``Curves'' in Lexicon Technicum by John Harris published in London in 1710. Newton also classified all cubics into 72 types, missing six of them. In addition, he showed that any cubic can be obtained by a suitable projection of the Elliptic Curve

$$y^2 = ax^3 + bx^2 + cx + d,$$

where the projection is a Birational Transformation, and the general cubic can also be written as

$$y^2 = x^3 + ax + b.$$

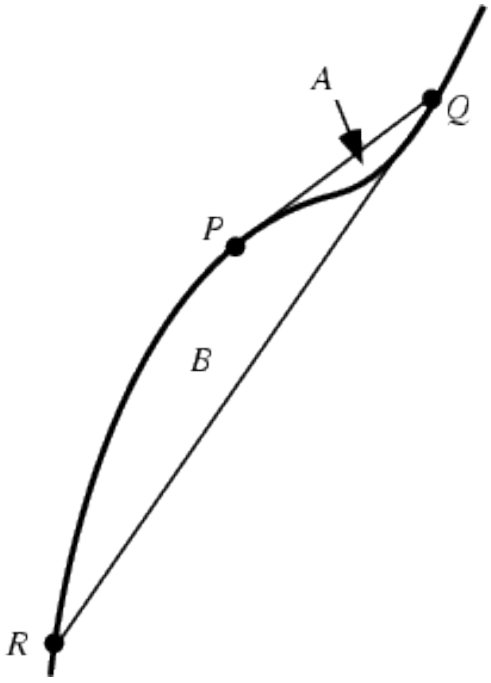
Newton's first class is equations of the form

$$xy^2 + ey = ax^3 + bx^2 + cx + d.$$

This is the hardest case and includes the Serpentine Curve as one of the subcases. The third class was

$$ay^2 = x(x^2 - 2bx + c),$$

which is called Newton's Diverging Parabolas. Newton's 66th curve was the Trident of Newton. Newton's classification of cubics was criticized by Euler because it lacked generality. Pucker later gave a more detailed classification with 219 types.



Pick a point P, and draw the tangent to the curve at P. Call the point where this tangent intersects the curve Q. Draw another tangent and call the point of intersection with the curve R. Every curve of third degree has the property that, with the areas in the above labeled figure,

$$B = 16A$$

Quadric Surface

Quadric surfaces are defined by quadratic equations in two dimensional space. Spheres

and cones are examples of quadrics. The quadric surfaces of RenderMan are surfaces of revolution in which a finite curve in two dimensions is swept in three dimensional space about one axis to create a surface. A circle centered at the origin forms a sphere. If the circle is not centered at the origin, the circle sweeps out a torus. A line segment with one end lying on the axis of rotation forms a cone. A line segment parallel to the axis of rotation forms a cylinder. The generalization of a line segment creates a hyperboloid by rotating an arbitrary line segment in three dimensional space about the Z axis. The axis of rotation is always the z axis. Each quadric routine has a sweep parameter, specifying the angular extent to which the quadric is swept about z axis. Sweeping a quadric by less than 360 degrees leaves an open surface.

Quadrics

Many common shapes can be modeled with quadrics. Although it is possible to convert quadrics to patches, they are defined as primitives because special-purpose rendering programs render them directly and because their surface parameters are not necessarily preserved if they are converted to patches. Quadric primitives are particularly useful in solid and molecular modeling applications.

All the following quadrics are rotationally symmetric about the z axis. In all the quadrics u and v are assumed to run from 0 to 1. These primitives all define a bounded region on a quadric surface. It is not possible to define infinite quadrics. Note that each quadric is defined relative to the origin of the object coordinate system. To position them at another point or with their symmetry axis in another direction requires the use a modeling transformation. The geometric normal to the surface points ``outward'' from the z-axis, if the current orientation matches the orientation of the current transformation and "inward" if they don't match. The sense of a quadric can be reversed by giving negative parameters. For example, giving a negative thetamax parameter in any of the following definitions will turn the quadric inside-out.

Each quadric has a parameter list. This is a list of token-array pairs where each token is one of the standard geometric primitive variables or a variable which has been defined with Rideshare. Position variables should not be given with quadrics. All angular arguments to these functions are given in degrees. The trigonometric functions used in their definitions are assumed to also accept angles in degrees.

Solid Modeling

A solid model is a digital representation of the geometry of an existing or envisioned physical object. Solid models are used in many industries, from entertainment to health care. They play a major role in the discrete-part manufacturing industries, where precise models of parts and assemblies are created using solid modeling software or more

general computer-aided design (CAD) systems. The design process is usually incremental. Designers may specify points, curves, and surfaces, and stitch them together to define electronic representations of the boundary of the object. Alternatively, they may select models of simple shapes, such as blocks or cylinders, specify their dimensions, position, and orientation, and combine them using union, intersection, or difference operators. The resulting representation is an unambiguous, complete, and detailed digital approximation of the geometry of an object or of an assembly of objects (such as a car engine or an entire airplane). Interactive three-dimensional (3D) graphic supports the design activities by providing designers with: (1) easy to understand images of their design, (2) efficient facilities for graphically selecting or editing features of the part being designed, and (3) immediate feedback, which helps them perform and check each design step.

Early applications of solid modeling focused on producing correct engineering drawings automatically and on cutter-path generation for numerically controlled machining [Requicha82, Requicha96]. Today, the engineering drawings still provide a link to traditional, non-electronic manufacturing or archival processes, but are being rapidly replaced with electronic file transfers. Solid modeling has evolved to provide the set of fundamental tools for representing a large class of products and processes, and for performing on them the geometric calculations required by applications. The ultimate goal is to relieve engineers from all of the low-level or non-creative tasks in designing products; in assessing their manufacturability, assemblability, and other life-cycle characteristics; and in generating all the information necessary to produce them. Engineers should be able to focus on conceptual, high-level design decisions, while domain-expert application programs should provide advice on the consequences of design decisions and generate plans for the manufacture and other activities associated with the product's life cycle. The total automation of analysis and manufacturing activities (see for example [Spyridi93]), although in principle made possible by informationally complete solid models, remains a research challenge in spite of much progress on several fronts.

Solid modeling has rapidly evolved into a large body of knowledge, created by an explosion of research and publications [Requicha88, Requicha92]. The solid modeling technology is implemented in dozens of commercial solid modeling software systems, which serve a multi-billion dollar market and have significantly increased design productivity, improved product quality, and reduced manufacturing and maintenance costs.

Today, solid modeling is an interdisciplinary field that involves a growing number of areas. Its objectives evolved from a deep understanding of the practices and

requirements of the targeted application domains. Its formulation and rigor are based on mathematical foundations derived from general and algebraic topology, and from Euclidean, differential, and algebraic geometry. The computational aspects of solid modeling deal with efficient data structures and algorithms, and benefit from recent developments in the field of computational geometry. Efficient processing is essential, because the complexity of industrial models is growing faster than the performance of commercial workstations. Techniques for modeling and analyzing surfaces and for computing their intersections are important in solid modeling. This area of research, sometimes called computer aided geometric design, has strong ties with numerical analysis and differential geometry. Graphic user-interface (GUI) techniques also play a crucial role in solid modeling, since they determine the overall usability of the modeler and impact the user's productivity. There have always been strong symbiotic links and overlaps between the solid modeling community and the computer graphics community. Solid modeling interfaces are based on efficient three-dimensional (3D) graphics techniques, whereas research in 3D graphics focuses on fast or photo-realistic rendering of complex scenes, often composed of solid models, and on realistic or artistic animations of non-rigid objects. A similar symbiotic relation with computer vision is regaining popularity, as many research efforts in vision are model-based and attempt to extract 3D models from images or video sequences of existing parts or scenes. These efforts are particularly important for solid modeling, because the cost of manually designing solid models of existing objects or scenes far exceeds the other costs (hardware, software, maintenance, and training) associated with solid modeling. Finally, the growing complexity of solid models and the growing need for collaboration, reusability of design, and interoperability of software require expertise in distributed databases, constraint management systems, optimization techniques, object linking standards, and internet protocols.

Solid modeling systems

A solid modeling system, often called a solid modeler, is a computer program that provides facilities for storing and manipulating data structures that represent the geometry of individual objects or assemblies. These representations can be created either by a human through a graphic user interface (GUI), or specified by software applications via an application programming interface (API). In a typical interactive application, a user selects and manipulates modeling primitives (parameterized instances of simple geometric shapes, such as a cylinder or a line) and invokes modeling operations that combine or transform these primitives into more elaborate representations.

A modeler's GUI generates 3D graphic feedback to a user by immediately displaying selected portions of the objects being designed. In addition, it provides facilities for selecting and for graphically editing the displayed entities. Users of modern modelers can describe objects in terms of features, which are higher-level entities meaningful for their applications, can use dimensions and other constraints to help in sizing and positioning geometric entities, and can also parameterize the objects so as to create object families.

The choice of representations used by the modeler determines its domain (i.e., which objects can be modeled), and has a strong impact on the complexity and performance of the algorithms that create or process the representations. A modeler may support several distinct representation schemes. Consistency between representations in different schemes is typically enforced by representation conversion algorithms.

Application programs invoked by the users analyze the models and generate information on the object's properties (e.g., its moments of inertia, or its deflection under load, calculated by finite element methods), or on processes needed by life-cycle activities such as manufacture, assembly or inspection. In a modern engineering environment, these applications should run concurrently with the design process, to help assess the consequences of design decisions, and provide guidance to the designers.

Representing Solids

Most geometric objects we see every day are solids. That is, they are geometric objects with interior. Solids can be very simple like a cube or very complex like a piston engine. To be processed by computers, solids must have some representations that can describe the geometry and characteristics completely. In fact, a good representation should address the following issues:

1. **Domain** :
   While no representation can describe all possible solids, a representation should be able to represent a useful set of geometric objects.

2. **Unambiguity** :
   When you see a representation of a solid, you will know what is being represented without any doubt. An unambiguous representation is usually referred to as a complete one.

3. **Uniqueness** :
   That is, there is only one way to represent a particular solid. If a representation is unique, then it is easy to determine if two solids are identical since one can just compare their representations.

4. **Accuracy** :
   A representation is said accurate if no approximation is required.

5. **Validness** :
   This means a representation should not create any invalid or impossible solids. More precisely, a representation will not represent an object that does not correspond to a solid.

6. **Closure** :
   Solids will be transformed and used with other operations such as union and intersection. "Closure" means that transforming a valid solid always yields a valid solid.

7. **Compactness and Efficiency** :
   A good representation should be compact enough for saving space and allow for efficient algorithms to determine desired physical characteristics.

These issues may be contradictory with each other. For efficiency purpose, a curvilinear solid may be approximated by a polyhedron. There are many efficient and robust algorithms for handling polyhedra; however, accuracy may not be maintained in the process of approximation. For example, given two curvilinear solids that are tangent to each other, this tangency may disappear after converting to a polyhedron.

Problems occur even for the polyhedra world. Many graphics APIs such as PHIGS PLUS and OpenGL have built-in data structures for representing polyhedra; but, these representations could generate invalid solids. There are representations that can always represent valid solids; but, these representations are in general more complex than those available in graphics APIs.

In summary, designing representations for solids is a difficult job and compromises are often necessary. This course will only discuss the following representations: wireframes, boundary representations and constructive solid geometry.

Regularized Boolean Set Operation primitive Instancing Sweep Representations

A way to show the physical object via mathematical operation or logical operation in solid modeling.

Terms and definition:

Definition:

1. A regularized Boolean set-operation op can be obtained by first taking the interior of the resultant point set of an ordinary Boolean setoperation (Pop Q) and then by taking the closure That is, Pop Q = closure(interior (Pop Q)). Regularized Boolean set-operations appear in Constructive Solid Geometry (CSG), because regular sets are closed under regularized Boolean set-operations, and because regularization eliminates lower dimensional features, namely isolated vertices and antennas, thus simplifying and restricting the representation to physically meaningful solids.

2. The counterclockwise cyclic sequence of alternating polygon edges and polygon vertices is referred to as the polygon (outer) boundary.

3. A Relatively simple polygon allows vertices with a degree>2, but all of its edges are disjoint in their interior. Furthermore, it must be an orient able polygon. Namely when it is inserted into an arrangement and its outer boundary is traversed, the same face is adjacent to all of the half edges (no crossing over any curve during the traversal). Note that while polygon C has the same curves as polygon B, traversal of the curves leads to crossing over a previously traversed curve, and is therefore neither simple nor relatively simple.

4. A polygon in our context must be relatively simple and its outer boundary vertices must be ordered in a counterclockwise direction around the interior of the polygon. We extend the notion of a polygon to a point set in $\mathbb{R}$ that has a topology of a polygon and its boundary edges must map to xmonotone curves, and refer to it as a general polygon. We sometimes use the term polygon instead of general polygon for simplicity hereafter.

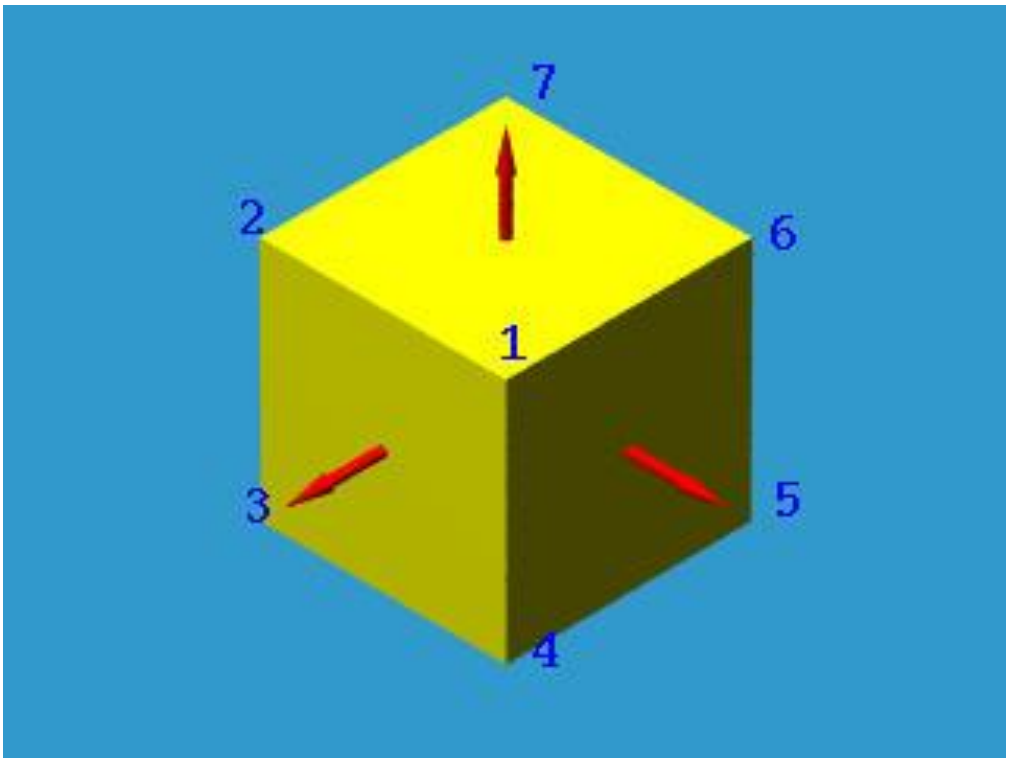
Boundary Representations

Boundary Representation, or B-rep for short, can be considered as an extension to the wireframe model. The merit of a B-rep is that a solid is bounded by its surface and has its interior and exterior. The surface of a solid consists of a set of well-organized faces, each of which is a piece of some surface (.e.g., a surface patch). Faces may share

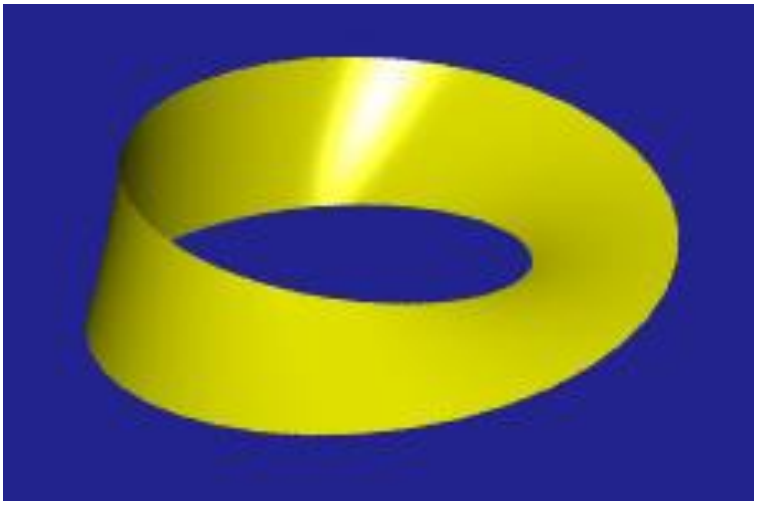
vertices and edges that are curve segments. Therefore, a B-rep is an extension to the wireframe model by adding face information to the latter.

There are two types of information in a B-rep: topological and geometric. Topological information provide the relationships among vertices, edges and faces similar to that used in a wireframe model. In addition to connectivity, topological information also include orientation of edges and faces. Geometric information are usually equations of the edges and faces.

The orientation of each face is important. Normally, a face is surrounded by a set of vertices. Using the right-handed rule, the ordering of these vertices for describing a particular face must guarantee that the normal vector of that face is pointing to the exterior of the solid. Normally, the order is counter clockwise. If that face is given by an equation, the equation must be rewritten so that the normal vector at every point on the part that is being used as a face points to the exterior of the solid. Therefore, by inspecting normal vectors one can immediately tell the inside and outside of a solid under B-rep. This orientation must be done for all faces. The following shows three faces and their outward pointing normal vectors. To describe the top surface, the vertices should be 6, 7, 2, 1 or 7, 2, 1, 6 or 2, 1, 6, 7 or 1, 6, 7, 2. To describe the left face, the order should be 1, 2, 3, 4 or 2, 3, 4, 1 or 3, 4, 1, 2 or 4, 1, 2, 3.



Unfortunately, not all surfaces can be oriented this way. If the surface of a solid can be oriented this way, it is called orientable; otherwise, it is non-orientable. The following shows the well-known Mobius band which is one-sided and non-orientable.



Constructive Solid Geometry Comparison of Representations

Constructive Solid Geometry, or CSG for short, is yet another way of representing solids. A CSG solid is constructed from a few primitives with Boolean operators (i.e., set union, intersection and difference). Thus, a CSG solid can be written as a set equations and can also be considered a design methodology.

CSG Primitives

The standard CSG primitives consist of the block (i.e., cube), triangular prism, sphere, cylinder, cone and torus. These six primitives are in some normal or generic form and must be instantiated by the user to be used in his/her design. Moreover, the instantiated primitive may require transformations such as scaling, translation and rotation to be positioned at the desired place.

Suppose the block primitive is defined by its "lower left" corner < -1, -1, -1 > and "upper right" corner < 1, 1, 1 >. To produce a rectangular box with center at < 3, 2, 3 > and height and width 3 and length 5, a user may first scale the block primitive 1.5 times in the y- and z-direction and 2.5 times in the x-direction, and then translate the result to < 3, 2, 3 >. If the block primitive is called Block in a CSG system, the result may be obtained as follows:

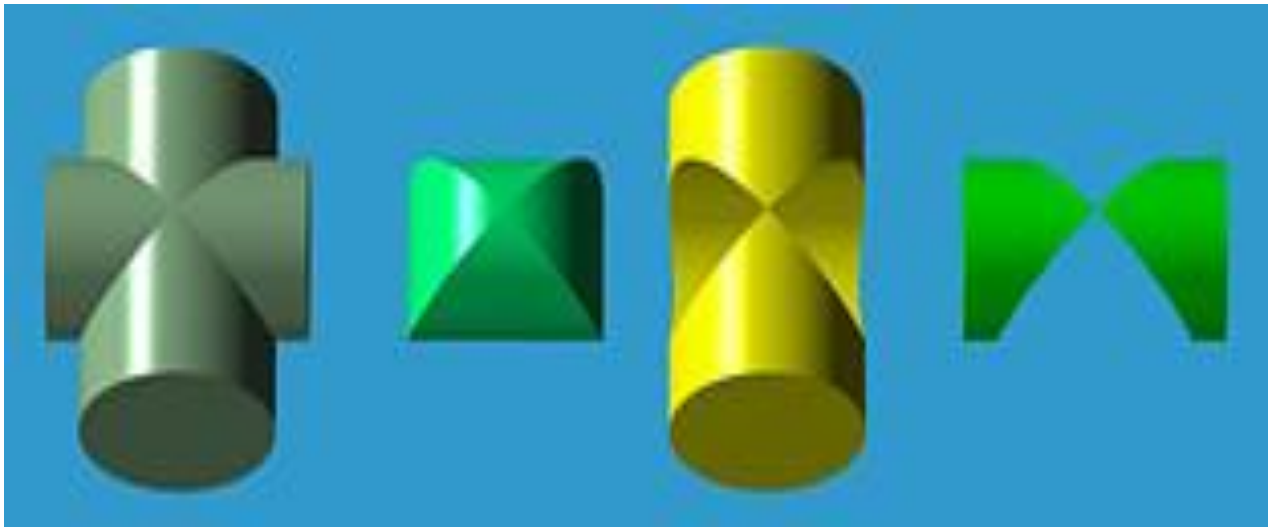
translate(scale(Block, < 2.5, 1.5, 1.5 >), < 3, 2, 3 >)

In the above, the object to be transformed and the transformation data are the first and second arguments, respectively.

Boolean Operators

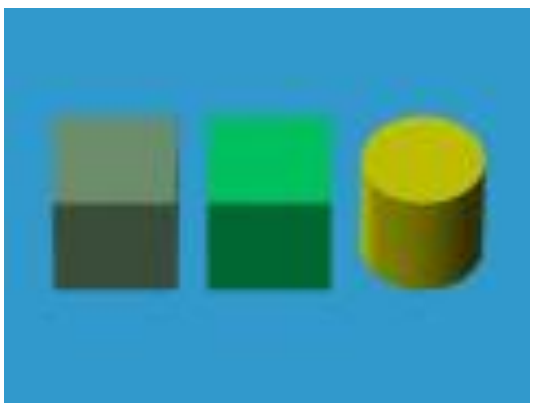
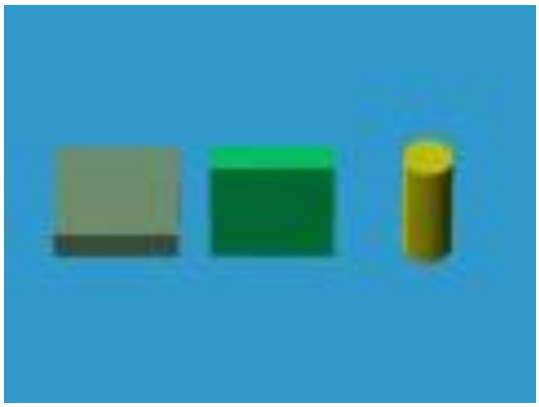
We can combined two instantiated and perhaps transformed primitives into one with set union, set intersection and set difference operators. However, simple set operators may create problems as will be discussed in regularized Boolean operators, modifications are required. Let us just use set operations on this page.

Given two sets, A and B, its union consists of all points from either A or B; its intersection consists of all points in both sets; and its difference, written as A - B (resp., B - A), consists of all points in A but not in B (resp., in B but not in A). In the following, A is the vertical cylinder and B is the horizontal cylinder. From left to right, the four solids are the union and intersection of A and B, A - B and B - A.



Therefore, a solid can be considered as the result of applying Boolean operators to a set of instantiated and transformed CSG primitives.

Let us take a look at a simple example. We want to design a bracket-like shape with a hole shown on the right-most figure below. We start with two instantiations of blocks and one instantiation of a cylinder (the left-most figure). Then, the two blocks are scaled and one of them is rotated to a vertical position. The cylinder is also scaled so that its radius matches that of the hole. These three instantiations are than transformed to their desired positions. The final product is obtained by computing the union of the two blocks and then subtracting from it the cylinder.

Please note that the design of the above solid is not unique. For example, the L shape can be constructed from subtracting a cube from another one.

CSG Expressions

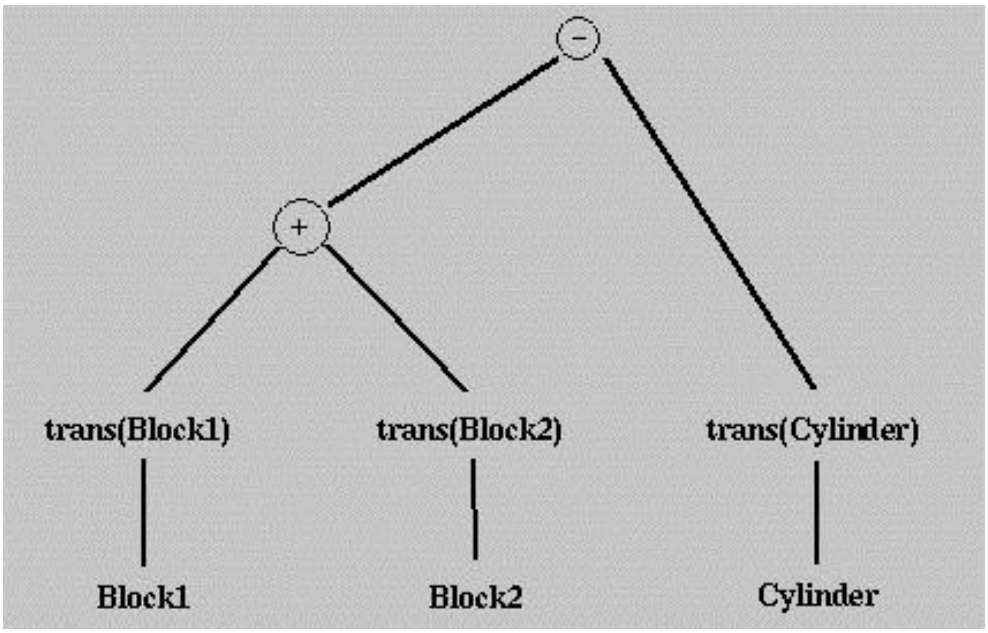
The design procedure of the above bracket can be written as an expression:

diff(union(trans(Block1), trans(Block2)), trans(Cylinder))

where union(A,B) and diff(A,B) are the union and difference of A and B,
and trans() indicates appropriate transformations. Or, if we use +, ^ and - for set union, intersection and difference, the above function calls can be rewritten as a set expression as follows:

(trans(Block1) + trans(Block2)) - trans(Cylinder)

This expression can be converted to an expression tree, the CSG Expression, of the design:



In fact, every solid constructed using the CSG technique has a corresponding CSG expression which in turn has an associated CSG tree. The expression of the CSG tree is a representation of the final design. Recall that the same solid may have different CSG expressions/trees. For example, one might punch a hole from Block1 first and then compute the union of this result with Block2. As a result, CSG representations are not unique.

UNIT-V

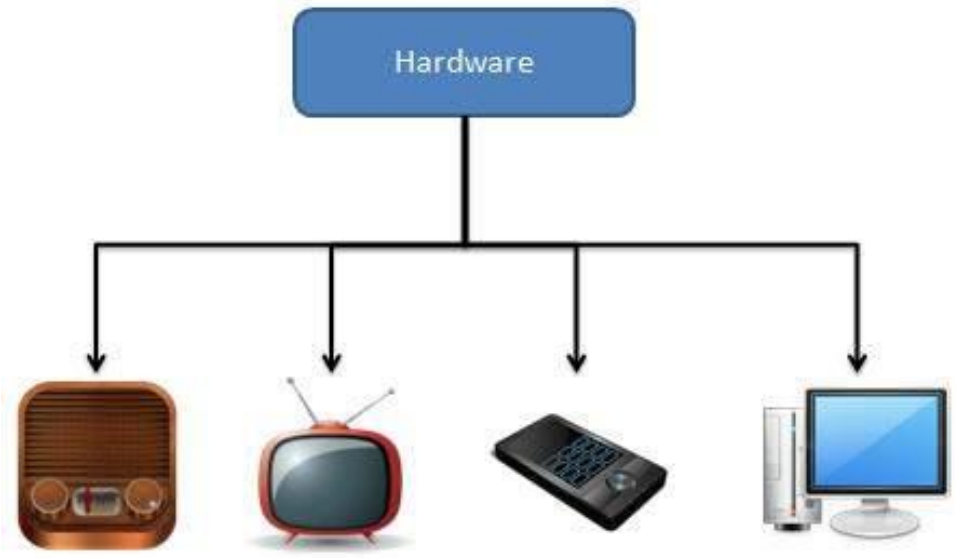
Introductory Concepts:
Multimedia Definition

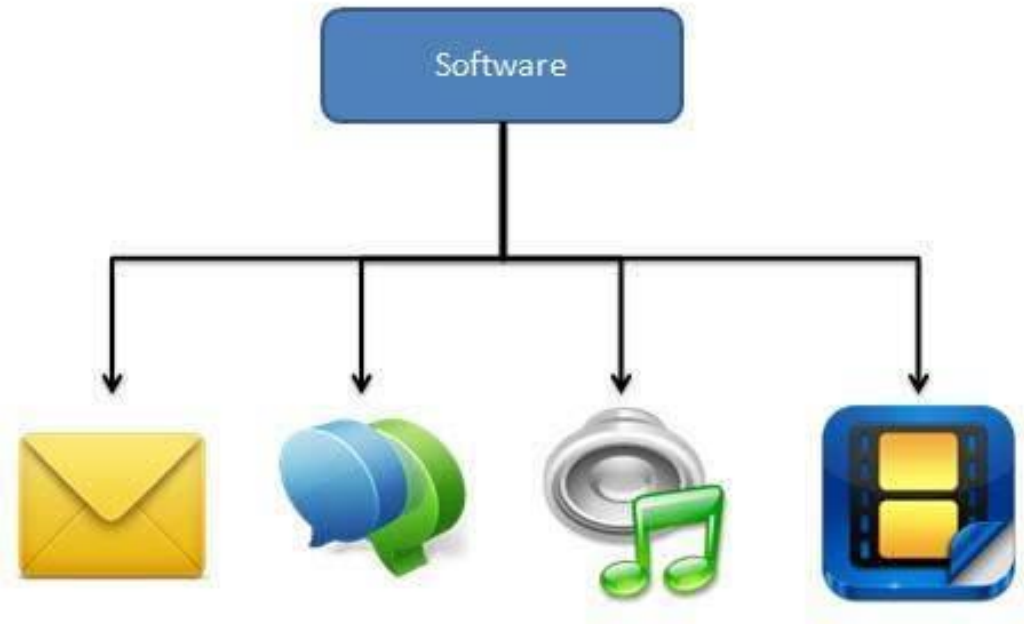
Multimedia is an interactive media and provides multiple ways to represent information to the user in a powerful manner. It provides an interaction between users and digital information. It is a medium of communication.Some of the sectors where multimedias is used extensively are education, training, reference material, business presentations, advertising and documentaries.

Definition of Multimedia

By definition Multimedia is a representation of information in an attractive and interactive manner with the use of a combination of text, audio, video, graphics and animation. In other words we can say that Multimedia is a computerized method of presenting information combining textual data, audio, visuals (video), graphics and animations. For examples: E-Mail, Yahoo Messenger, Video Conferencing, and Multimedia Message Service (MMS).

Multimedia as name suggests is the combination of Multi and Media that is many types of media (hardware/software) used for communication of information.





Components of Multimedia

Following are the common components of multimedia:

- **Text**- All multimedia productions contain some amount of text. The text can have various types of fonts and sizes to suit the profession presentation of the multimedia software.
- **Graphics**- Graphics make the multimedia application attractive. In many cases people do not like reading large amount of textual matter on the screen. Therefore, graphics are used more often than text to explain a concept, present background information etc. There are two types of Graphics:
    - **Bitmap images**- Bitmap images are real images that can be captured from devices such as digital cameras or scanners. Generally bitmap images are not editable. Bitmap images require a large amount of memory.
    - **Vector Graphics**- Vector graphics are drawn on the computer and only require a small amount of memory. These graphics are editable.
- **Audio**- A multimedia application may require the use of speech, music and sound effects. These are called audio or sound element of multimedia.Speech is also a perfect way for teaching. Audio are of analog and digital types. Analog audio or sound refers to the original sound signal. Computer stores the sound in digital form. Therefore, the sound used in multimedia application is digital audio.

- **Video**- The term video refers to the moving picture, accompanied by sound such as a picture in television. Video element of multimedia application gives a lot of information in small duration of time. Digital video is useful in multimedia application for showing real life objects. Video have highest performance demand on the computer memory and on the bandwidth if placed on the internet. Digital video files can be stored like any other files in the computer and the quality of the video can still be maintained. The digital video files can be transferred within a computer network. The digital video clips can be edited easily.
- **Animation**- Animation is a process of making a static image look like it is moving. An animation is just a continuous series of still images that are displayed in a sequence. The animation can be used effectively for attracting attention. Animation also makes a presentation light and attractive. Animation is very popular in multimedia application

Applications of Multimedia

Following are the common areas of applications of multimedia.

- **Multimedia in Business**- Multimedia can be used in many applications in a business. The multimedia technology along with communication technology has opened the door for information of global wok groups. Today the team members may be working anywhere and can work for various companies. Thus the work place will become global. The multimedia network should support the following facilities:
    - Voice Mail
    - Electronic Mail
    - Multimedia based FAX
    - Office Needs
    - Employee Training
    - Sales and Other types of Group Presentation
    - Records Management
- **Multimedia in Marketing and Advertising**- By using multimedia marketing of new products can be greatly enhanced. Multimedia boost communication on an affordable cost opened the way for the marketing and advertising personnel. Presentation that have flying banners, video transitions, animations, and sound effects are some of the elements used in composing a multimedia based

advertisement to appeal to the consumer in a way never used before and promote the sale of the products.

- **Multimedia in Entertainment**- By using multimedia marketing of new products can be greatly enhanced. Multimedia boost communication on an affordable cost opened the way for the marketing and advertising personnel. Presentation that have flying banners, video transitions, animations, and sound effects are some of the elements used in composing a multimedia based advertisement to appeal to the consumer in a way never used before and promote the sale of the products.
- **Multimedia in Education**- Many computer games with focus on education are now available. Consider an example of an educational game which plays various rhymes for kids. The child can paint the pictures, increase reduce size of various objects etc apart from just playing the rhymes.Several other multimedia packages are available in the market which provide a lot of detailed information and playing capabilities to kids.
- **Multimedia in Bank**- Bank is another public place where multimedia is finding more and more application in recent times. People go to bank to open saving/current accounts, deposit funds, withdraw money, know various financial schemes of the bank, obtain loans etc. Every bank has a lot of information which it wants to impart to in customers. For this purpose, it can use multimedia in many ways. Bank also displays information about its various schemes on a PC monitor placed in the rest area for customers. Today on-line and internet banking have become very popular. These use multimedia extensively. Multimedia is thus helping banks give service to their customers and also in educating them about banks attractive finance schemes.
- **Multimedia in Hospital**- Multimedia best use in hospitals is for real time monitoring of conditions of patients in critical illness or accident. The conditions are displayed continuously on a computer screen and can alert the doctor/nurse on duty if any changes are observed on the screen. Multimedia makes it possible to consult a surgeon or an expert who can watch an ongoing surgery line on his PC monitor and give online advice at any crucial juncture.

  In hospitals multimedia can also be used to diagnose an illness with CD-ROMs/ Cassettes/ DVDs full of multimedia based information about various diseases and their treatment.Some hospitals extensively use multimedia presentations in training their junior staff of doctors and nurses. Multimedia displays are now extensively used during critical surgeries.
- **Multimedia Pedagogues**- Pedagogues are useful teaching aids only if they stimulate and motivate the students. The audio-visual support to a pedagogue

can actually help in doing so. A multimedia tutor can provide multiple numbers of challenges to the student to stimulate his interest in a topic. The instruction provided by pedagogue have moved beyond providing only button level control to intelligent simulations, dynamic creation of links, composition and collaboration and system testing of the user interactions.

- **Communication Technology and Multimedia Services**- The advancement of high computing abilities, communication ways and relevant standards has started the beginning of an era where you will be provided with multimedia facilities at home. These services may include:
    - Basic Television Services
    - Interactive entertainment
    - Digital Audio
    - Video on demand
    - Home shopping
    - Financial Transactions
    - Interactive multiplayer or single player games
    - Digital multimedia libraries
    - E-Newspapers, e-magazines

CD-ROM and the multimedia highway

Multimedia require large amount t of digital memory when stores in an end users library, or large amounts of bandwidth when distributed over wires, glass Fiber, or airwaves on network. The greater the bandwidth, the bigger the "pipeline", so more content can be delivered to end users quickly.

CD-ROM, DVD and Multimedia:

CD-ROM (compact disc read-only memory, has become the most cost-effective distribution medium for multimedia projects: a CD-ROM disc can be mass-produced for pennies and can contain up to 80 minutes of full-screen video or sound. Or it can come rain unique mixes of images, sound, text, video and animations controlled by an authoring system to provide ultimates user interaction.

Discs can be stamped out of poly-carbonate plastic as fast as cookies on a baker's production line and just as cheaply. Virtually all personal computers sold today include a least a CD-ROM player, and the software that drives these computers is commonly available on a CD-ROM disc applications that required inserting as many as 16 or more floppy disk one after another are now installed from a CD-ROM without muss or fuss.

Many systems now come with a DVD-ROM player, Multilayered Digital Versatile Disk (DVD) technology increases the capacity and multimedia capability or current optical technology to 18 GB. CD and DVD burners are used for reading discs and for making them, too, in audio, video, and data formats. DVD authoring and integration software allows the creation of interactive front-end menus for films and games.

In the very long term, however, CD-ROM and DVD discs are but interim memory technologies that will be replaced by new devices that do not require moving parts. As the data highway described below becomes more and more pervasive and users become better "connected", copper wire, glass Fiber, and radio/cellular technologies may prevail as the most common delivery means for interactive multimedia files, served across the broadband internet or from dedicated computer farms and storage facilities.

The Multimedia Highway:

Now, that telecommunications networks are global, and when information provides and content owners determines the worth of theirs products and how to charge money for them, information elements will ultimately link up online's as distributed resources on a data highway (actually more like a toll road). Where you will pay to acquire and use multimedia-based information.

Curiously, the actual glass Fiber cables that makes up much of the physical backbone of the data highway are, in many cases. Owned by railroad and pipelines companies who simply buried the cables on existing rights of way where no special permits and

environmental reports are necessary. One railroad in the United States invested more than a million dollars in a special cable laying trenching car; in the United Kingdom, there is talk of placing a fiber-optic cables backbone along the decaying 19th century canal and barge system. Bandwidth on these lines is leased to other, so competing retailers such as AT&T, MCI, and Sprint may even share the same cable. Full-text content from books and magazines is accessible by modem and electronic link; features movies are played at home; real-time new reports from anywhere on earth are available; lectures from participating universities are monitored for education credits; street maps of any city are view-able with recommendations for restaurants, in any language-and online travelogues include testimonials and video tracks. This is not science fiction; it is happing now. For each of these interfaces or gateways to information is a Multimedia projects just waiting to be developed.

Computer Animation (Design, types of animation, using different functions) Uses of Multimedia

Animation

Animation refers to the movement on the screen of the display device created by displaying a sequence of still images. Animation is the technique of designing, drawing, making layouts and preparation of photographic series which are integrated into the multimedia and gaming products. Animation connects the exploitation and management of still images to generate the illusion of movement. A person who creates animations is called animator. He/she use various computer technologies to capture the pictures and then to animate these in the desired sequence.

Animation includes all the visual changes on the screen of display devices. These are:

1. Change of shape as shown in fig:

Fig: Change in Shape

2. Change in size as shown in fig:

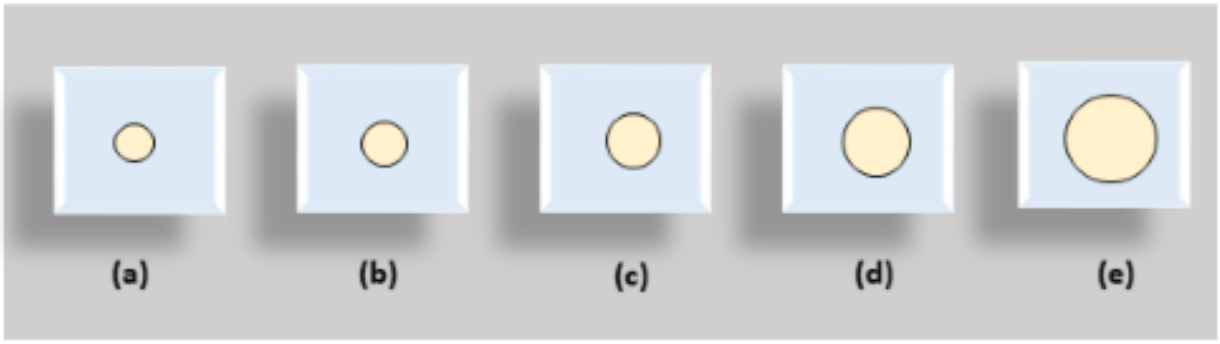


Fig: Change in Size

3. Change in color as shown in fig:

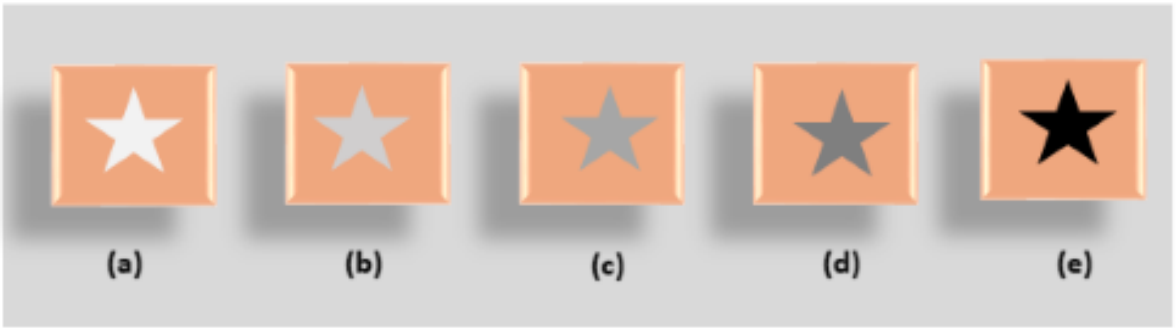


Fig: Change in Color

4. Change in structure as shown in fig:

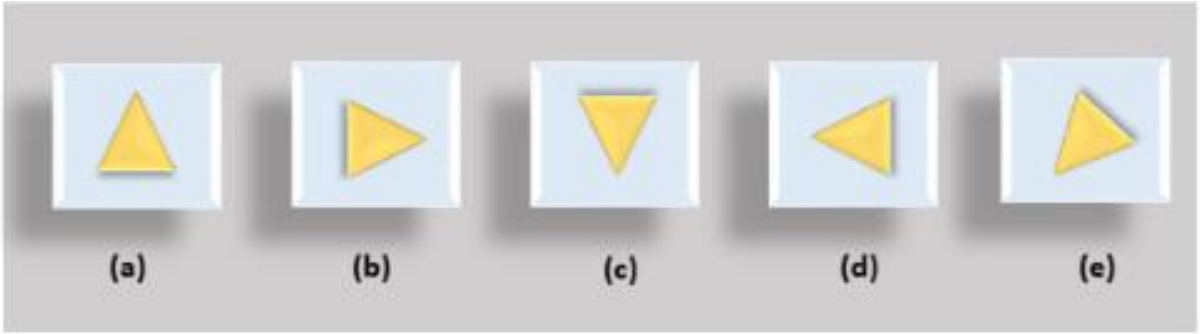


Fig: Change in Structure

5. Change in angle as shown in fig:

Animation means giving life to any object in computer graphics. It has the power of injecting energy and emotions into the most seemingly inanimate objects. Computer-assisted animation and computer-generated animation are two categories of computer animation. It can be presented via film or video.

The basic idea behind animation is to play back the recorded images at the rates fast enough to fool the human eye into interpreting them as continuous motion. Animation can make a series of dead images come alive. Animation can be used in many areas like entertainment, computer aided-design, scientific visualization, training, education, e-commerce, and computer art.

Animation Techniques

Animators have invented and used a variety of different animation techniques. Basically there are six animation technique which we would discuss one by one in this section.

Traditional Animation framebyframe
Traditionally most of the animation was done by hand. All the frames in an animation had to be drawn by hand. Since each second of animation requires 24 frames filmfilm, the amount of efforts required to create even the shortest of movies can be tremendous.

Keyframing

In this technique, a storyboard is laid out and then the artists draw the major frames of the animation. Major frames are the ones in which prominent changes take place. They are the key points of animation. Keyframing requires that the animator specifies critical or key positions for the objects. The computer then automatically fills in the missing frames by smoothly interpolating between those positions.

Procedural

In a procedural animation, the objects are animated by a procedure − a set of rules − not by keyframing. The animator specifies rules and initial conditions and runs simulation. Rules are often based on physical rules of the real world expressed by mathematical equations.

Behavioral

In behavioral animation, an autonomous character determines its own actions, at least to a certain extent. This gives the character some ability to improvise, and frees the animator from the need to specify each detail of every character's motion.

Performance Based MotionCaptureMotionCapture

Another technique is Motion Capture, in which magnetic or vision-based sensors record the actions of a human or animal object in three dimensions. A computer then uses these data to animate the object.

This technology has enabled a number of famous athletes to supply the actions for characters in sports video games. Motion capture is pretty popular with the animators mainly because some of the commonplace human actions can be captured with relative ease. However, there can be serious discrepancies between the shapes or dimensions of the subject and the graphical character and this may lead to problems of exact execution.

Physically Based DynamicsDynamics

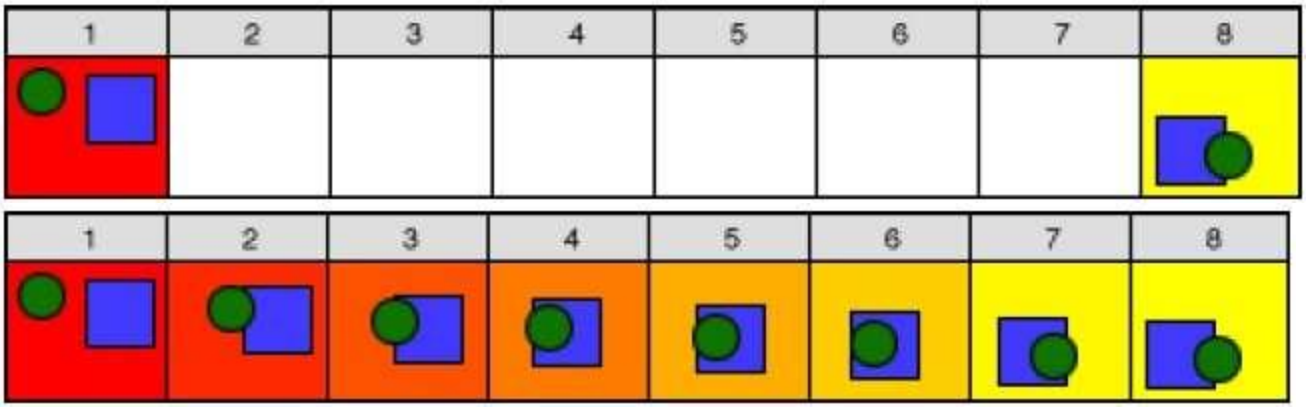
Unlike key framing and motion picture, simulation uses the laws of physics to generate motion of pictures and other objects. Simulations can be easily used to produce slightly different sequences while maintaining physical realism. Secondly, real-time simulations allow a higher degree of interactivity where the real person can maneuver the actions of the simulated character.

In contrast the applications based on key-framing and motion select and modify motions form a pre-computed library of motions. One drawback that simulation suffers from is the expertise and time required to handcraft the appropriate controls systems.

Key Framing

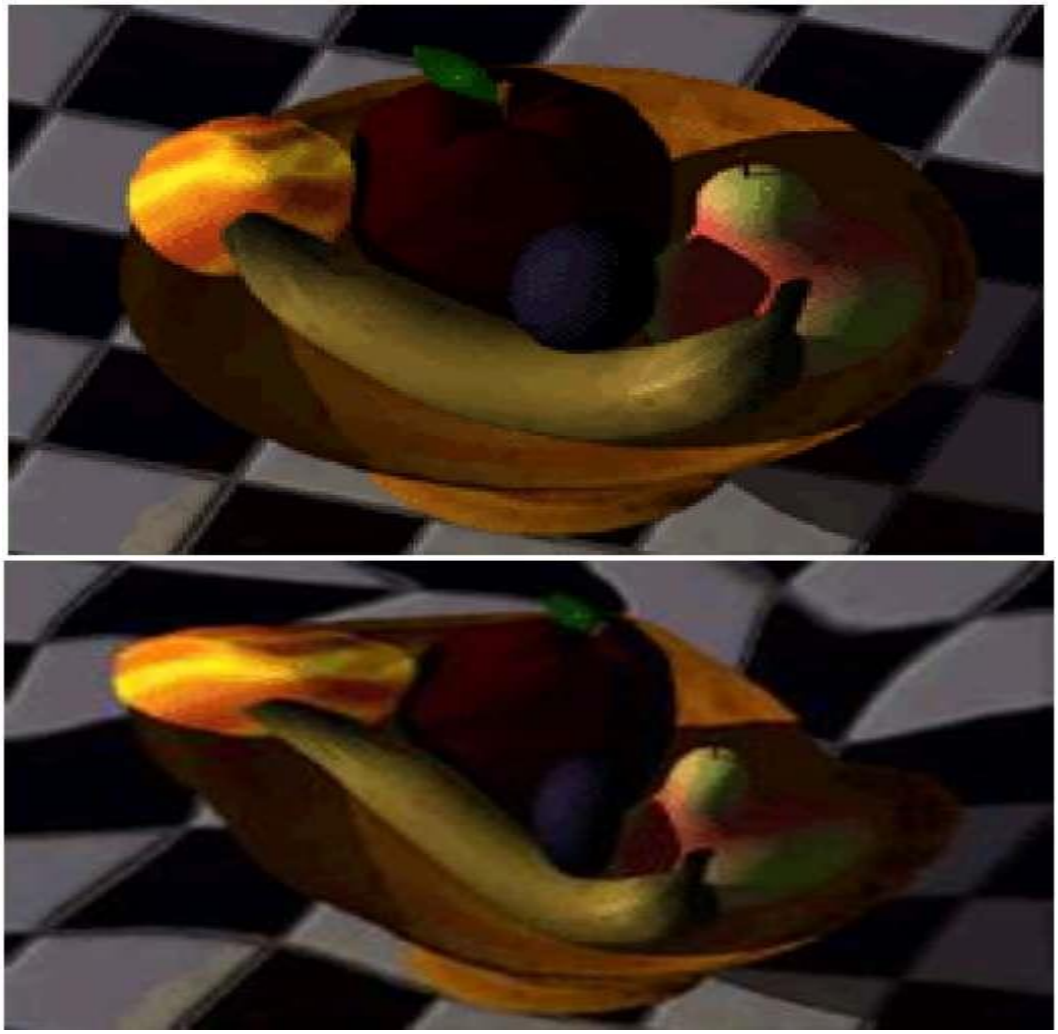
A keyframe is a frame where we define changes in animation. Every frame is a keyframe when we create frame by frame animation. When someone creates a 3D animation on a computer, they usually don’t specify the exact position of any given object on every single frame. They create keyframes.

Keyframes are important frames during which an object changes its size, direction, shape or other properties. The computer then figures out all the in-between frames and saves an extreme amount of time for the animator. The following illustrations depict the frames drawn by user and the frames generated by computer.



Morphing

The transformation of object shapes from one form to another form is called morphing. It is one of the most complicated transformations.



A morph looks as if two images melt into each other with a very fluid motion. In technical terms, two images are distorted and a fade occurs between them.

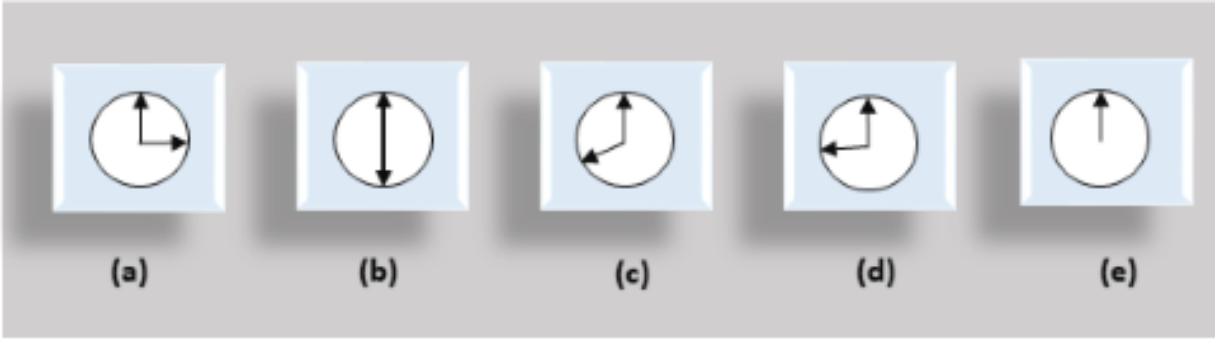


Fig: Change in angle

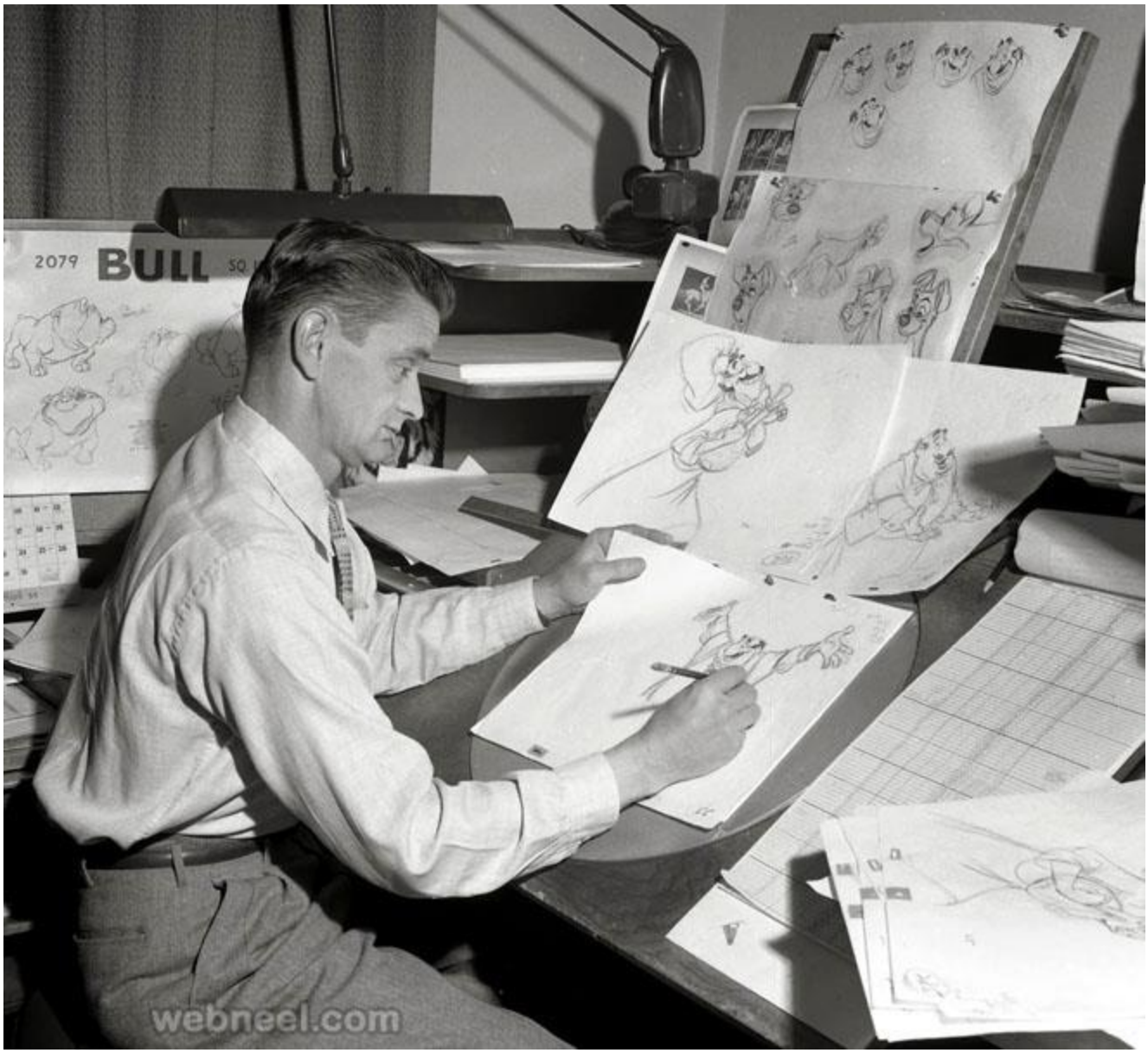
Types Of Animaiton

In this post we listed 20 different types of animation techniques and styles. Animation is the process of creating the illusion of motion and shape change by means of the rapid display of a sequence of static images that minimally differ from each other. Animation is all around us, be it your favourite tv commercials, music, movies or even videos you can see the stop motion animation type. Movement creation techniques incorporate the conventional traditional animation and stop motion animation techniques of two and three-dimensional figures, for example, paper set patterns, puppets and clay figures. Keeping Stop motion as the base of all animation, different styles of animation techniques can be used to create the animated sequences. In this post we included 20 different types of animaiton and animtion styles,

1. Traditional animation
2. 2D animation
3. 3D animation

Traditional animation

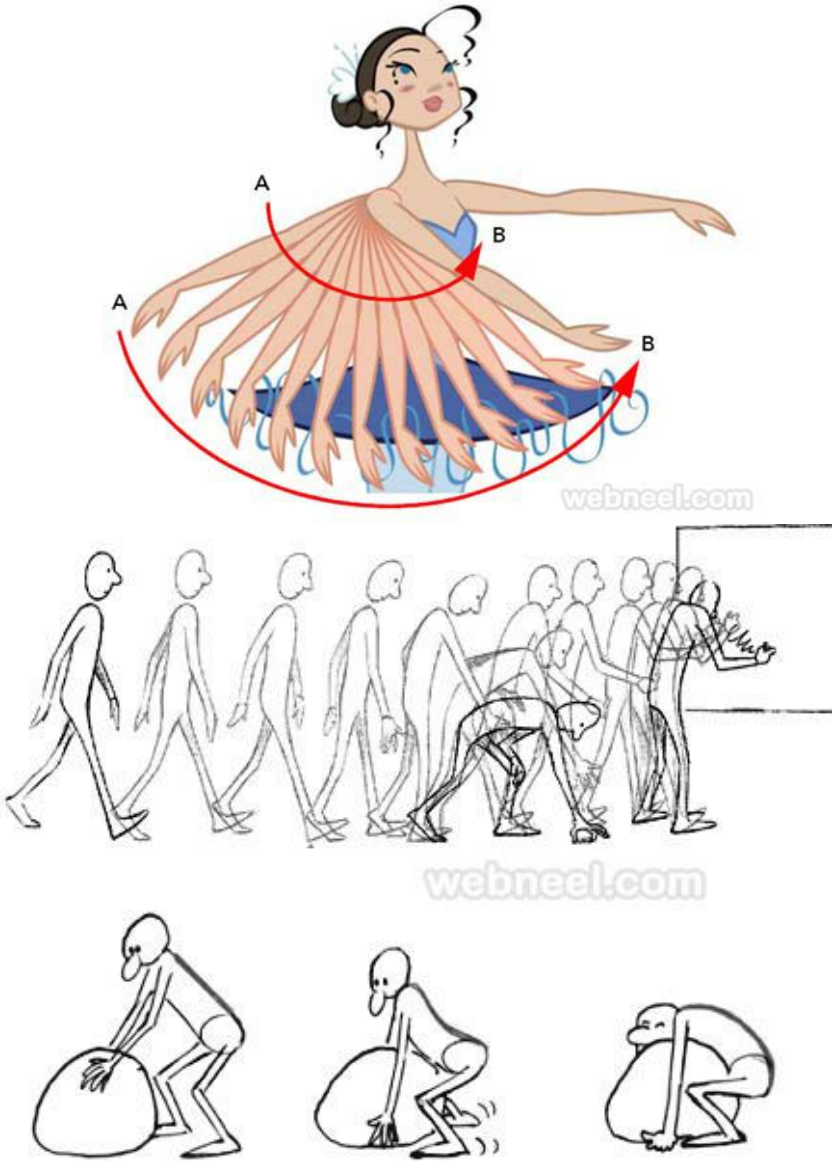
Traditional animation involved animators drawing by hand for each and every frame. If you love the feel of pencils on a paper, then the traditional approach is very fascinating. Traditional animation is creating the drawings one by one on the frame. 2D animation involves creating numerous drawings then feeding into a plastic cells, hand painting them and create the animated sequence on a painted background image.
Traditional Aniamtion Movies : Snow White and the Seven Dwarfs, Peter Pan, and Sleeping Beauty, Aladdin



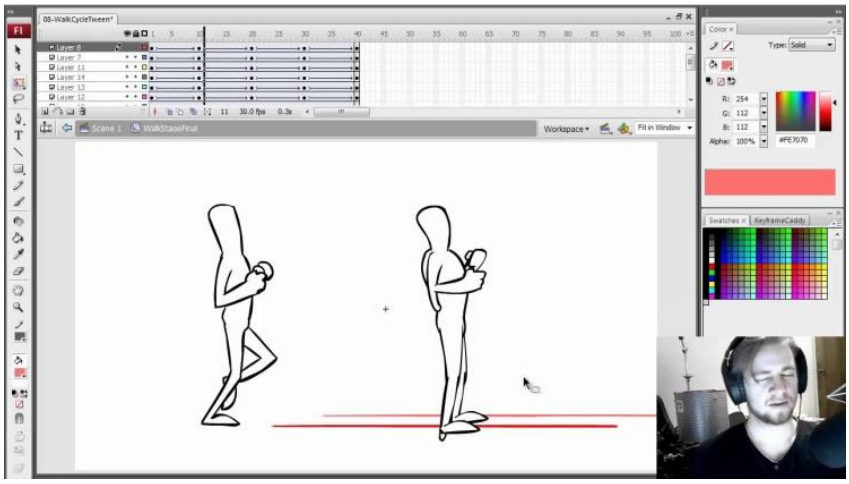
Computer animation - 2D, 3D: The famous Mickey Mouse animation was created using the 2d animation technique.The first 2D animation was called Fantasmagorie, it's a short cartoon made by Emile Cohl. It's shot entirely in black and white, the cartoon is all about a simple stick man in live action. The cartoon is 75 seconds long and it took about 700 different drawings to create. This historic animation was released in 1908. During the 1960s many popular cartoons like the Jetsons and the Flintstones were created using 2d animation.

2D animation

Creating animations in the 2 dimensional space with the help of digital technologies is known as digital 2d animation. You don't need to create digital models, you just need to draw the frames. Create 100s of drawing and animating them to show some kind of movement is technically known as digital 2d animation. Using Adobe flash, animators can limit the number of drawings used, which makes them easier to create digital 2d animation. Small variations like changing the color or frame rate can be changed almost instantly, thus making it easier for the animators to work on.



Simple 2D walk cyle : Watch Video



3D animation

If you are interested in making the unreal characters into a realistic one, then it's Digital 3d animation. Digital 3d animation characters are much faster to create and they are quite popular in the movie making industry. Using a computer software 3d animated images are used to create many short films, full length movies and even tv commercials and a career in digital 3d animation is highly rewarding. Comparing to 2D animation and the traditional approach, 3d animation models are highly realistic.

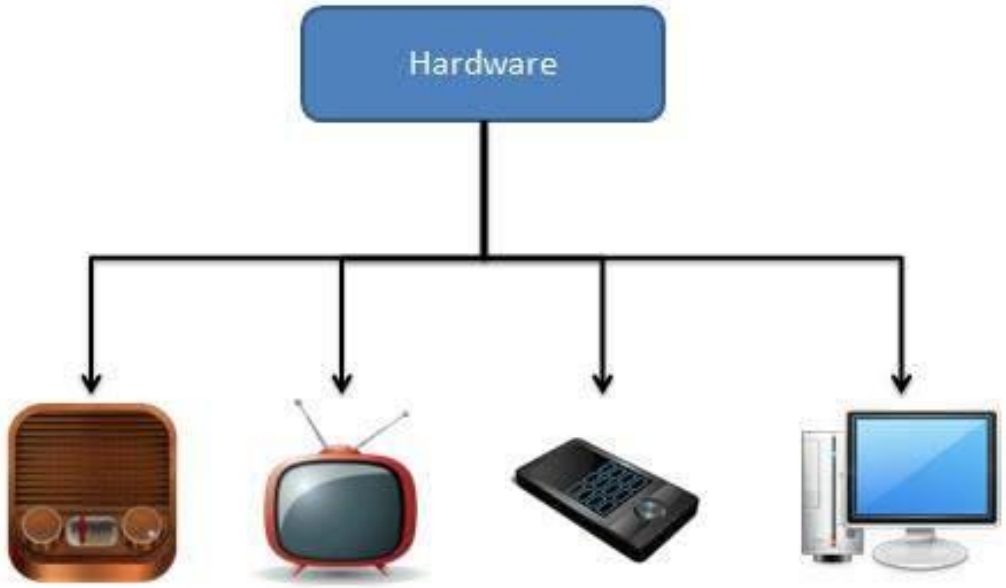
Introduction to making multimedia – The stage of Project

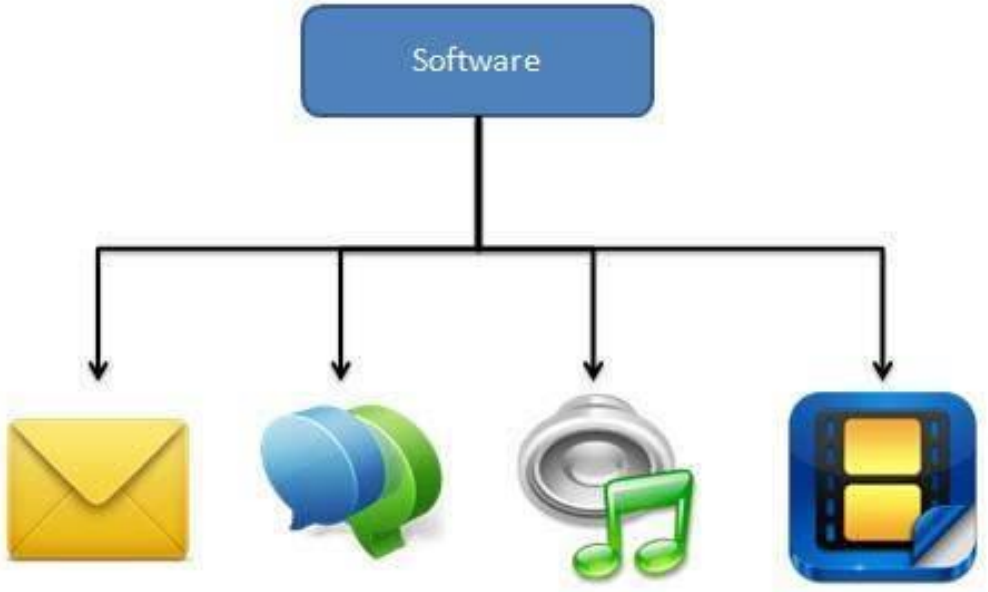
Multimedia is an interactive media and provides multiple ways to represent information to the user in a powerful manner. It provides an interaction between users and digital information. It is a medium of communication.Some of the sectors where multimedias is used extensively are education, training, reference material, business presentations, advertising and documentaries.

Definition of Multimedia

By definition Multimedia is a representation of information in an attractive and interactive manner with the use of a combination of text, audio, video, graphics and animation. In other words we can say that Multimedia is a computerized method of presenting information combining textual data, audio, visuals (video), graphics and animations. For examples: E-Mail, Yahoo Messenger, Video Conferencing, and Multimedia Message Service (MMS).

Multimedia as name suggests is the combination of Multi and Media that is many types of media (hardware/software) used for communication of information.





Components of Multimedia

Following are the common components of multimedia:

- **Text**- All multimedia productions contain some amount of text. The text can have various types of fonts and sizes to suit the profession presentation of the multimedia software.

- **Graphics**- Graphics make the multimedia application attractive. In many cases people do not like reading large amount of textual matter on the screen. Therefore, graphics are used more often than text to explain a concept, present background information etc. There are two types of Graphics:
    - **Bitmap images**- Bitmap images are real images that can be captured from devices such as digital cameras or scanners. Generally bitmap images are not editable. Bitmap images require a large amount of memory.
    - **Vector Graphics**- Vector graphics are drawn on the computer and only require a small amount of memory. These graphics are editable.
- **Audio**- A multimedia application may require the use of speech, music and sound effects. These are called audio or sound element of multimedia.Speech is also a perfect way for teaching. Audio are of analog and digital types. Analog audio or sound refers to the original sound signal. Computer stores the sound in digital form. Therefore, the sound used in multimedia application is digital audio.
- **Video**- The term video refers to the moving picture, accompanied by sound such as a picture in television. Video element of multimedia application gives a lot of information in small duration of time. Digital video is useful in multimedia application for showing real life objects. Video have highest performance demand on the computer memory and on the bandwidth if placed on the internet. Digital video files can be stored like any other files in the computer and the quality of the video can still be maintained. The digital video files can be transferred within a computer network. The digital video clips can be edited easily.
- **Animation**- Animation is a process of making a static image look like it is moving. An animation is just a continuous series of still images that are displayed in a sequence. The animation can be used effectively for attracting attention. Animation also makes a presentation light and attractive. Animation is very popular in multimedia application

Applications of Multimedia

Following are the common areas of applications of multimedia.

- **Multimedia in Business**- Multimedia can be used in many applications in a business. The multimedia technology along with communication technology has opened the door for information of global wok groups. Today the team members may be working anywhere and can work for various companies. Thus the work

place will become global. The multimedia network should support the following facilities:

- Voice Mail
- Electronic Mail
- Multimedia based FAX
- Office Needs
- Employee Training
- Sales and Other types of Group Presentation
- Records Management

- **Multimedia in Marketing and Advertising**- By using multimedia marketing of new products can be greatly enhanced. Multimedia boost communication on an affordable cost opened the way for the marketing and advertising personnel. Presentation that have flying banners, video transitions, animations, and sound effects are some of the elements used in composing a multimedia based advertisement to appeal to the consumer in a way never used before and promote the sale of the products.
- **Multimedia in Entertainment**- By using multimedia marketing of new products can be greatly enhanced. Multimedia boost communication on an affordable cost opened the way for the marketing and advertising personnel. Presentation that have flying banners, video transitions, animations, and sound effects are some of the elements used in composing a multimedia based advertisement to appeal to the consumer in a way never used before and promote the sale of the products.
- **Multimedia in Education**- Many computer games with focus on education are now available. Consider an example of an educational game which plays various rhymes for kids. The child can paint the pictures, increase reduce size of various objects etc apart from just playing the rhymes.Several other multimedia packages are available in the market which provide a lot of detailed information and playing capabilities to kids.
- **Multimedia in Bank**- Bank is another public place where multimedia is finding more and more application in recent times. People go to bank to open saving/current accounts, deposit funds, withdraw money, know various financial schemes of the bank, obtain loans etc. Every bank has a lot of information which it wants to impart to in customers. For this purpose, it can use multimedia in many ways. Bank also displays information about its various schemes on a PC

monitor placed in the rest area for customers. Today on-line and internet banking have become very popular. These use multimedia extensively. Multimedia is thus helping banks give service to their customers and also in educating them about banks attractive finance schemes.

- **Multimedia in Hospital**- Multimedia best use in hospitals is for real time monitoring of conditions of patients in critical illness or accident. The conditions are displayed continuously on a computer screen and can alert the doctor/nurse on duty if any changes are observed on the screen. Multimedia makes it possible to consult a surgeon or an expert who can watch an ongoing surgery line on his PC monitor and give online advice at any crucial juncture.

  In hospitals multimedia can also be used to diagnose an illness with CD-ROMs/ Cassettes/ DVDs full of multimedia based information about various diseases and their treatment.Some hospitals extensively use multimedia presentations in training their junior staff of doctors and nurses. Multimedia displays are now extensively used during critical surgeries.

- **Multimedia Pedagogues**- Pedagogues are useful teaching aids only if they stimulate and motivate the students. The audio-visual support to a pedagogue can actually help in doing so. A multimedia tutor can provide multiple numbers of challenges to the student to stimulate his interest in a topic. The instruction provided by pedagogue have moved beyond providing only button level control to intelligent simulations, dynamic creation of links, composition and collaboration and system testing of the user interactions.

- **Communication Technology and Multimedia Services**- The advancement of high computing abilities, communication ways and relevant standards has started the beginning of an era where you will be provided with multimedia facilities at home. These services may include:
  - Basic Television Services
  - Interactive entertainment
  - Digital Audio
  - Video on demand
  - Home shopping
  - Financial Transactions
  - Interactive multiplayer or single player games
  - Digital multimedia libraries
  - E-Newspapers, e-magazines

hardware & software requirements to make good multimedia skills and Training opportunities in Multimedia Motivation for Multimedia usage

The computer technology that allows us to develop three-dimensional virtual environments (VEs) consists of both hardware and software. The current popular, technical, and scientific interest in VEs is inspired, in large part, by the advent and availability of increasingly powerful and affordable visually oriented, interactive, graphical display systems and techniques. Graphical image generation and display capabilities that were not previously widely available are now found on the desktops of many professionals and are finding their way into the home. The greater affordability and availability of these systems, coupled with more capable, single-person-oriented viewing and control devices (e.g., head-mounted displays and hand-controllers) and an increased orientation toward real-time interaction, have made these systems both more capable of being individualized and more appealing to individuals.

Limiting VE technology to primarily visual interactions, however, simply defines the technology as a more personal and affordable variant of classical military and commercial graphical simulation technology. A much more interesting, and potentially useful, way to view VEs is as a significant subset of multimodal user interfaces. Multimodal user interfaces are simply human-machine interfaces that actively or purposefully use interaction and display techniques in multiple sensory modalities (e.g., visual, haptic, and auditory). In this sense, VEs can be viewed as multimodal user interfaces that are interactive and spatially oriented. The human-machine interface hardware that includes visual and auditory displays as well as tracking and haptic interface devices is covered in Chapters

In this chapter, we focus on the computer technology for the generation of VEs. One possible organization of the computer technology for VEs is to decompose it into functional blocks. In three distinct classes of blocks are shown: (1) rendering hardware and software for driving modality-specific display devices; (2) hardware and software for modality-specific aspects of models and the generation of corresponding display representations; (3) the core hardware and software in which modality-independent aspects of models as well as consistency and registration among multimodal models are taken into consideration. Beginning from left to right, human sensorimotor systems, such as eyes, ears, touch, and speech, are connected to the computer through human-machine interface devices. These devices generate output to, or receive input from, the human as a function of sensory modal drivers or renderers. The auditory display driver, for example, generates an appropriate waveform based on an acoustic simulation of the VE. To generate the sensory output, a computer must simulate the VE for that particular sensory mode. For example, a haptic display may require a physical simulation that includes compliance and texture. An acoustic display may require sound models based on impact, vibration, friction, fluid flow, etc. Each sensory modality requires a simulation

tailored to its particular output. Next, a unified representation is necessary to coordinate individual sensory models and to synchronize output for each sensory driver. This representation must account for all human participants in the VE, as well as all autonomous internal entities. Finally, gathered and computed information must be summarized and broadcast over the network in order to maintain a consistent distributed simulated environment.

To date much of the design emphasis in VE systems has been dictated by the constraints imposed by generating the visual scene. The nonvisual modalities have been relegated to special-purpose peripheral devices. Similarly, this chapter is primarily concerned with the visual domain, and material on other modalities can be found in Chapters 3-7. However, many of the issues involved in the modeling and generation of acoustic and haptic images are similar to the visual domain; the implementation requirements for interacting, navigating, and communicating in a virtual world are common to all modalities. Such multimodal issues will no doubt tend to be merged into a more unitary computational system as the technology advances over time.

In this section, we focus on the computer technology for the generation of VEs. The computer hardware used to develop three-dimensional VEs includes high-performance workstations with special components for multisensory displays, parallel processors for the rapid computation of world models, and high-speed computer networks for transferring information among participants in the VE. The implementation of the virtual world is accomplished with software for interaction, navigation, modeling (geometric, physical, and behavioral), communication, and hypermedia integration. Control devices and head-mounted displays are covered elsewhere in this report.

VE requires high frame rates and fast response because of its inherently interactive nature. The concept of frame rate comes from motion picture technology. In a motion picture presentation, each frame is really a still photograph. If a new photograph replaces the older images in quick succession, the illusion of motion in engendered. The update rate is defined to be the rate at which display changes are made and shown on the screen. In keeping with the original motion picture technology, the ideal update rate is 20 frames (new pictures) per second or higher. The minimum acceptable rate for VE is lower, reflecting the trade-offs between cost and such tolerances. With regard to computer hardware, there are several senses of frame rate: they are roughly classified as graphical, computational, and data access. Graphical frame rates are critical in order to sustain the illusion of presence or immersion in a VE. Note that these frame rates may be independent: the graphical scene may change without a new computation and data access due to the motion of the user's point of view. Experience has shown that, whereas the graphical frame rate should be as high as possible, frame rates of lower than 10 frames per second severely degrade the illusion of presence. If the graphics being displayed relies on computation or data access, then computation and data

access frame rates of 8 to 10 frames per second are necessary to sustain the visual illusion that the user is watching the time evolution of the VE.

Fast response times are required if the application allows interactive control. It is well known (Sheridan and Ferrell, 1974) that long response times (also called lag or pure delay) severely degrade user performance. These delays arise in the computer system from such factors as data access time, computation time, and rendering time, as well as from delays in processing data from the input devices. As in the case of frame rates, the sources of delay are classified into data access, computation, and graphical categories. Although delays are clearly related to frame rates, they are not the same: a system may have a high frame rate, but the image being displayed or the computational result being presented may be several frames old. Research has shown that delays of longer than a few milliseconds can measurably impact user performance, whereas delays of longer than a tenth of a second can have a severe impact. The frame rate and delay required to create a measurable impact will in general depend on the nature of the environment. Relatively static environments with slowly moving objects are usable with frame rates as low as 8 to 10 per s and delays of up to 0.1 s. Environments with objects exhibiting high frequencies of motion (such as a virtual handball game) will require very high frame rates (> 60 Hz) and very short delays. In all cases, however, if the frame rate falls below 8 frames per s, the sense of an animated three-dimensional environment begins to fail, and if delays become greater than 0.1 s, manipulation of the environment becomes very difficult. We summarize these results to the following constraints on the performance of a VE system:

Frame rates must be greater than 8 to 10 frames/s.

Total delay must be less than 0.1 s.

Both the graphics animation and the reaction of the environment to user actions require extensive data management, computation, graphics, and network resources. All operations that take place to support the environment must operate within the above time constraints. Although one can imagine a system that would have the graphics, computation, and communications capability to handle all environments, such a system is beyond current technology. For a long time to come, the technology necessary will generally be dependent on the application domain for which the VE is being built. Real-world simulation applications will be highly bound by the graphics and network protocols and by consistency issues; information visualization and scientific visualization applications will be bound by the computational performance and will involve issues of massive data management (Bryson and Levit, 1992; Ellis et al., 1991). Some applications, such as architectural visualization, will require photorealistic rendering; others, such as information display, will not. Thus the particular hardware and software required for VE implementation will depend on the application domain targeted. There

are some commonalities of hardware and software requirements, and it is those commonalities on which we focus in our examination of the state of the art of computer hardware and software for the construction of real-time, three-dimensional virtual environments.

HARDWARE FOR COMPUTER GRAPHICS

The ubiquity of computer graphics workstations capable of real-time, three-dimensional display at high frame rates is probably the key development behind the current push for VEs today. We have had flight simulators with significant graphics capability for years, but they have been expensive and not widely available. Even worse, they have not been readily programmable. Flight simulators are generally constructed with a specific purpose in mind, such as providing training for a particular military plane. Such simulators are microcoded and programmed in assembly language to reduce the total number of graphics and central processing unit cycles required. Systems programmed in this manner are difficult to change and maintain. Hardware upgrades for such systems are usually major undertakings with a small customer base. An even larger problem is that the software and hardware developed for such systems are generally proprietary, thus limiting the availability of the technology. The graphics workstation in the last 5 years has begun to supplant the special-purpose hardware of the flight simulator, and it has provided an entry pathway to the large numbers of people interested in developing three-dimensional VEs. The following section is a survey of computer graphics workstations and graphics hardware that are part of the VE development effort.

Notable Graphics Workstations and Graphics Hardware

Graphics performance is difficult to measure because of the widely varying complexity of visual scenes and the different hardware and software approaches to computing and displaying visual imagery. The most straightforward measure is given in terms of polygons/second, but this only gives a crude indication of the scene complexity that can be displayed at useful interactive update rates. Polygons are the most common building blocks for creating a graphic image. It has been said that visual reality is 80 million polygons per picture (Catmull et al., 1984). If we wish photorealistic VEs at 10 frames/s, this translates into 800 million polygons/s. There is no current graphics hardware that provides this, so we must make approximations at the moment. This means living with less detailed virtual worlds, perhaps via judicious use of hierarchical data structures (see the software section below) or off-loading some of the graphics requirements by utilizing available CPU resources instead.

For the foreseeable future, multiple processor workstations will be playing a role in off-loading graphics processing. Moreover, the world modeling components, the

communications components, and the other software components for creating virtual worlds also require significant CPU capacity. While we focus on graphics initially, it is important to note that it is the way world modeling effects picture change that is of ultimate importance.

Graphics Architectures for VE Rendering

This section describes the high-level computer architecture issues that determine the applicability of a graphics system to VE rendering. Two assumptions are made about the systems included in our discussion. First, they use a z-buffer (or depth buffer), for hidden surface elimination. A z-buffer stores the depth—or distance from the eye point—of the closest surface "seen" at that pixel. When a new surface is scan converted, the depth at each pixel is computed. If the new depth at a given pixel is closer to the eye point than the depth currently stored in the z-buffer at that pixel, then the new depth and intensity information are written into both the z-buffer and the frame buffer. Otherwise, the new information is discarded and the next pixel is examined. In this way, nearer objects always overwrite more distant objects, and when every object has been scan converted, all surfaces have been correctly ordered in depth. The second assumption for these graphic systems is that they use an application-programmable, general-purpose processor to cull the database. The result is to provide the rendering hardware with only the graphics primitives that are within the viewing volume (a perspective pyramid or parallel piped for perspective and parallel projections respectively). Both of these assumptions are valid for commercial graphics workstations and for the systems that have been designed by researchers at the University of North Carolina at Chapel Hill.