Syllabus

Subject Code	**15BCA311**
Subject Name	**Computer Graphics Multimedia & Animation**
Short Name	**CGMA**
Total Teaching periods	**88**
Total Credits	**4**

Prerequisites :
- Basic knowledge about computer system, its components and functions.

Objectives:
- To acquire the basic knowledge about Computer Graphics.
- To learn the basic knowledge about various algorithms of Computer Graphics
- To understand the various types of multimedia & animation and compression techniques.

Units	**Contents**	**Total Lectures**
I	**Introduction to Computer Graphics:** Introduction, history, characteristics, advantages & disadvantages, application, components. Adapters ,Coordinates representation, software standards, **Input devices**: keyboard, mouse, trackball, space ball, joysticks, image scanner, touch panel, light pen, and voice system, **File format**: GIF, JPEG, PNG, TIFF, MPEG	18
II	**Graphics Preemptive** **Display devices**: CRT, Raster Scan Display, Random Scan Display, Flat Panel Display, LCD. Virtual reality system, Raster Scan System, Random Scan System.**Output Primitives** : Points & line, DDA algorithm, Bresenham's line algorithm, circle generation algorithm, **Attribute**: line, curve, text, area-fill	18
III	**Transformation and Clipping** **Transformation** : Introduction, translation, scaling, rotation, composite transformation, matrix representation, homogeneous coordinates, The Viewing Pipeline, Viewing Coordinate Reference Frame **Clipping:** point, Cohen-Sutherland Line Clipping, curve, text, exterior.	18
IV	**Multimedia and Compression** **Multimedia:** Introduction, overview, multimedia & hypermedia, advantages, disadvantages, application, software tools: music, Sequencing & notation, digital audio. **Compression:** introduction, need, types, evaluating & visibility, simple compression techniques, transform coding techniques.	17
V	**Animation:** Introduction, history, design of animation sequences, application, advantages, disadvantages, traditional animation, computer animation, tweening, morphing, case study on flash.	17
	Text Books : 1. Hearn d and Baker M. P, "Computer graphics-c version", 2nd edition Pearson Education. 2. Rajiv Chopda ," Computer graphics " revised edition S. Chand 3. ze-Nilan li, Mark S. Derw "Fundamental of multimedia"	
	1. Siamon J. Gibb And Dianysios c. Tsichritzis, "Multimedia Programming",AddisonWesely, 1995. 2. Johan Villain, Casanova And LeonyFernanadez, Eliar, "Multimedia Graphics", PHI, 1998 3. Malay K. Pakhira, Computer Graphics, Multimedia and Animation, 2ndEdition, PHI Publication.	

2 CGMA

Chapter Outline

Graphics Preemptive:-

Display devices:-
CRT
Raster Scan Display
Random Scan Display
Flat Panel Display
LCD
Virtual reality system
Raster Scan System
Random Scan System

Output Primitives:-
Points & Line
DDA Algorithm
Bresenham's line Algorithm
Circle generation Algorithm

Attribute:-
Line
Curve
Text
Area-fill

BASICS

Computer graphics is a sub-field of computer science and is concerned with digitally synthesizing and manipulating visual content. Although the term often refers to three-dimensional computer graphics, it also encompasses two-dimensional graphics and image processing.

Today, we find computer graphics used routinely in such diverse areas as science, engineering, medicine,**business,** industry, government, art, entertainment, advertising, education, and training.

COMPUTER AIDED DESIGN

A major **use** of computer graphics is in design processes, particularly for engineering and architectural systems, but almost all products are now computer designed. Generally referred to as CAD, computer-aided design methods are now routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft, computers, textiles, and many, many other products.

PRESENTATION GRAPHICS

Another major application **area** is presentation graphics, used to produce illustrations for **reports** or to generate **35-mm** slides or transparencies for use with projectors. Presentation graphics is commonly **used** to summarize financial, statistical, mathematical, scientific, and economic data for research reports, managerial reports, consumer information bulletins, and other types of reports. Workstation devices and **service bureaus**exist for converting screen displays into **35-mm** slides or overhead **transparencies** for **use** in presentations.

Typical examples of presentation graphics are bar charts, line graphs, surface graphs, pie **charts,** and other displays showing relationships between multiple parameters.

COMPUTER ART

Computer graphics methods are widely used in both fine art and commercial art applications. Artists use a variety of computer methods, including special-purpose hardware, artist's paintbrush (such as Lumens), other paint packages (such as Pixelpaint and Superpaint),**.**

VISUALIZATION

Scientists, engineers, medical personnel, business analysts, and others often need to analyze large amounts of information or to study the behavior of certain processes. Numerical simulations carried out on supercomputers frequently produce data files containing thousands and even millions of data values..

GRAPHICAL USER INTERFACES

It is common now for software packages to provide a graphical interface. A major component of a graphical interface is a window manager that allows a user to display multiple-window areas. Each window can contain a different process that can contain graphical or non-graphical displays. To make a particular window active, we simply click in that window using an interactive pointing device interfaces also display menus and icons for fast selection of processing options or parameter values.

1 What do you mean by display devices?

- Display devices are the interacting machines, which respond the users actions with the help of graphical images, animations, sounds etc. These devices are called as output devices, which show input as well as output.
- The major display devices are the screen or monitor or visual display units.
- Depending on the sizes, uses and even clarity, the devices are categorized in the different categories.

What are the different types of display devices?

The different types of display devices are as follows:

- **Refresh cathode ray tube technology**
- **Raster scan display**
- **Random scan display**
- **Flat panel displays**
- **Plasma panels**

1.1 Explain basis operation of CRT.

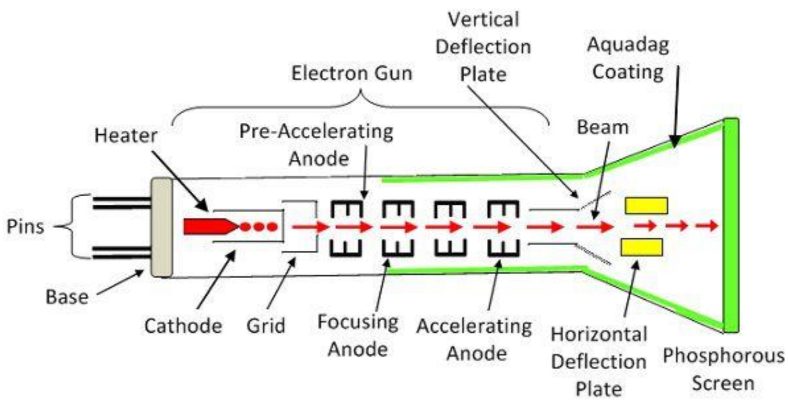
Basic operation of CRT:

- A beam of electron (cathode ray) is emitted by an electron gun, which passes through focusing and deflection system that directs the beam towards specified positions on the phosphor coated screen.
- The phosphor then emits a small spot of light at each position contacted by the electron beam.
- The light emitted by the phosphor fades very rapidly.
- To maintain the screen picture , one way is to redraw the picture repeatedly by quickly directing
  the electron beam back over the same points. Such a type of display is called refresh CRT.

Explain the working of Cathode Ray Tube (CRT).

- The electron beam passes between two pairs of metal plates, one vertical and other horizontal. A voltage difference is applied to each pair of plates according to the amount that the beam is to be deflected in each direction.
- In figure, the beam is first deflected towards one side of the screen. Then, as the beam passes through the horizontal plates, it is deflected towards the top or bottom of the screen.

- To get the proper deflection, adjust the current through coils placed around the outside of the CRT envelop.



Cathode Ray Tube

- The basic component of electron gun is heated metal, cathode and a control grid. as in figure, heating filament supplies heat, directing a current through the filament causes electrons to be boiled "off" the hot cathode surface.
- In the vacuum inside the CRT envelop, the free negatively charged metal coating on the on the side of CRT envelope near the phosphor screen or an accelerating anode can be used.

1.2 Write short note on Bean Penetration method.

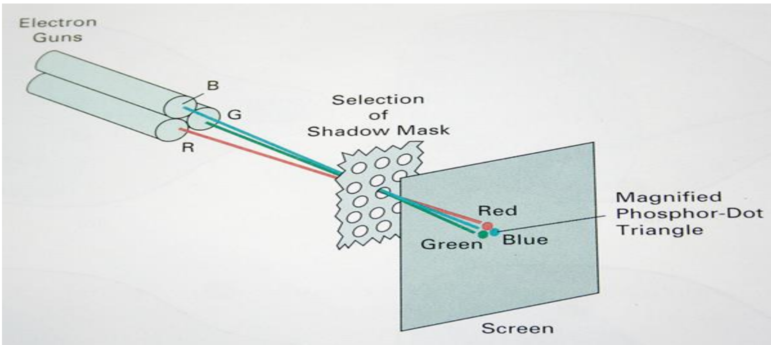
- Beam penetration method is used with random scan monitors.
- Two layer of phosphor (red and green) are coated onto the inside of CRT screen, displayed color depends on how far the electron beam penetrated into the phosphor layers.
- A beam of slow electrons excites only the outer red layer.
- A beam of very fast electron penetrates through the red layer and excites the inner green layer.
- At intermediate beam speeds, combinations, of red and green light are emitted to show two additional colors, orange and yellow.
- The speed of electrons is controlled by the beam acceleration voltage.
- Beam penetration method produces only four colors, and the quality of picture is not as good as with other methods.
- This method is an expensive way to produce colors in random scan monitors.

1.3 Draw a labeled diagram of shadow mask CRT.

OR

Describe shadow mask CRT with diagram.

- Shadow mask method is used in raster-scan systems because they produce a much wider range of colors.
- Shadow mask CRT has three phosphor color dots at each pixel position. One phosphor dot emits a red light, another emits a green light and the third emits a blue light.
- CRT has three electron guns, one for each color dot, and a shadow mask grid just behind the phosphor coated screen.
- The three electron beams are deflected and focused as group onto the shadow mask, containing a series of holes aligned with the phosphor dot patterns.
- Phosphor dots, are arranged in triangle so that each electron beam can activate only its corresponding color dot when it passes through the shadow mask.
- Different color combines are obtained by variant intensity levels of electrons beam.



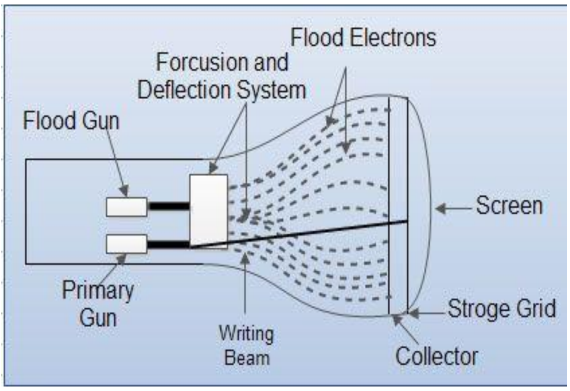
1.4 Describe working of Direct View Storage Tube (DVST) along with diagram.

Or

Explain Working principle of DVST.

- A direct view storage tube (DVST) stores the picture information as charge distribution just behind phosphor coated screen.
- The electron gun are used one is primary and second is flood gun. First one stores picture pattern and second one maintains picture display.
- The primary electron gun is used to draw the picture definition on the storage grid, a non-conducting material.
- High speed electron from primary gun strikes storage grid and knocks out electrons, which are attached to the collector grid.
- Storage grid being non-conducting , the areas where electrons have been removed will keep a net positive charge.
- This stored positive charge pattern on the storage grid is the picture definition.

- The flood gun produces a continuous stream of low speed electrons that pass through the control grid and are attached the positive areas of the storage grid.
- The low speed electrons penetrate through the storage grid to a the phosphor waiting, without affecting the charge pattern on the storage surface.



List the advantages and disadvantages of DVST (Direct View Storage Tube).

Advantages of DVST are as follows:

- No refreshing of CRT is required.
- The complex picture can be displayed at high resolution without flicker.
- Small DVST has a flat screen.

Disadvantages of DVST are as follows:

- They cannot displayed colors.
- Selective erasing of picture is not possible.
- It has poor contrast due to slow speed flood electron.
- For elimination of picture section redrawing has to be done.

.

1.5 Describing working principle of flat panel display.

- The flat display refers to a class of video device that have reduced volume, weight and power requirement compared to a CRT.
- A significant feature of flat panel display is that they are thinner than CRTs.
- Current uses for flat panel displays include small TV monitors, calculators, pocket video games etc.
- Flat panel display can be classified into two categories: emissive display and non-emissive display.
- The emissive displays are displays are devices that convert electrical energy into light. Plasma panel's thin film electron luminescent displays, and light emitting are example of emissive displays.
- In flat CRT's electron beams are acceleration parallel to the screen, and then deflected 90^0 to the screen. But flat CRTs have proved to be failure.
- Non emissive display use optical effectives to convert sunlight or light from some other source into graphics patterns. LCD is an example of non emissive flat panel display.

Write down the advantages and disadvantages of flat panel display

Advantages of Flat Panel Display are:

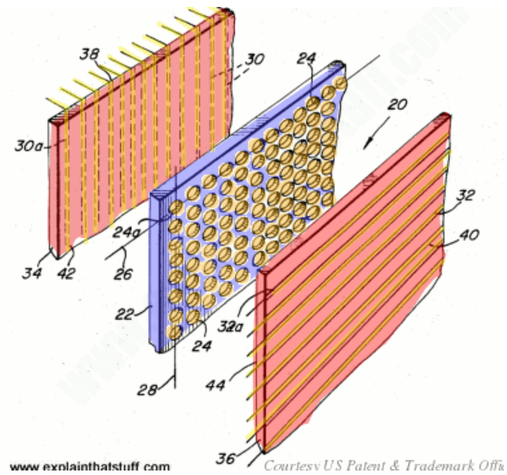
- It is light weight and less bulky device.
- It produces flicker free image.
- Refreshing is not required.
- It allows selecting, erasing and writing.

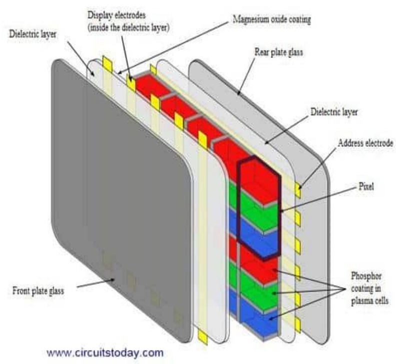
Disadvantages of Flat Panel Display are:

- It has poor resolution.
- It needs complex addressing and writing.
- It is very costly.

1.5.1 Describing working of Plasma Panel Display.

- Plasma panel display is also called gas discharge display. The plasma panel is an array of tiny neon bulbs.
- Each bulb can be put into “on” or off state, and remains in the state until explicitly changed to the other state. Plasma panels typically have 50to125 cells per inch.
- The neon bulbs are not discrete units, but rather are part of single integrated panel made of three layers of glass.
- The inside surface of the front layer has thin vertical strips of an electrical conductor. The center layer has number of holes (bulbs), and the inside surface of the real layer has thin horizontal strips of an electrical conductor.
- To turn on bulb, the system adjusts the voltages on the corresponding line such that their difference is large enough to put electrons from the neon molecules thus firing the bulb and making it glow.
- Once the glow starts, a lower voltage is applied to sustain it.
- To turn off, the system momentarily deceases the voltage on the appropriate lines to use the sustaining voltage. Bulbs can be turned on or off in about 15 seconds.

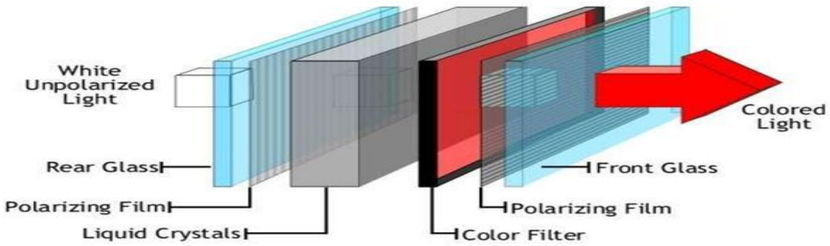




1.5.2 Write a note on Liquid Crystal Display(LCD).

- The term liquid crystal refers to the fact that these compounds have a crystalline arrangement of molecules, yet they flow like a liquid.
- Flat panel display commonly uses nematic (threadlike) liquefied crystal compounds that tend to keep the long axes of the rod shaped molecules aligned.
- In figure, two glass plates, sandwich the liquid crystal material. Row of horizontal transparent conductors are put into the other plate.
- The intersection of two conductors defines a pixel position. Normally the molecules are aligned as shown in the "On state".
- Polarized light passing through the material is twisted so that it will pass through the opposite polarizer.
- The light is then reflected back to the viewer. To turn off the pixel, apply a voltage to the two intersecting conductor to align the molecules do that light is not twisted.
- Picture definition is stored in a refresh buffer, and the screen is refreshed at the rate of 60 frames per second.

  Diagram



1.5.3 Write short notes on LED

LED stands for light emitting diode. You know those little lights on your computer, usually near the hard disk, that flash while the computer is working? Those are LEDs. They work on the principle of electroluminescence, which refers to substances that glow when you apply electricity. LEDs were used in digital watches, but now all digital watches use LCDs because LCD stakes less power.

LEDs are ordinary diodes (the most basic electronic component) that, due to their composition, happen to glow red, green, or amber when energized by a couple of volts. They use less power than incandescent bulbs and last over 100,000 hours.

1.6 Raster scan and Random scan

Q. Describe the difference between raster scan and random scan with respect to

(1) How it stores information in memory?

(2) How it displays stored information.

Q. Differentiate between random scan and raster scan display.

Q. What do you mean by the term raster scan?

Q. Write down the advantages and disadvantages of random scan display.

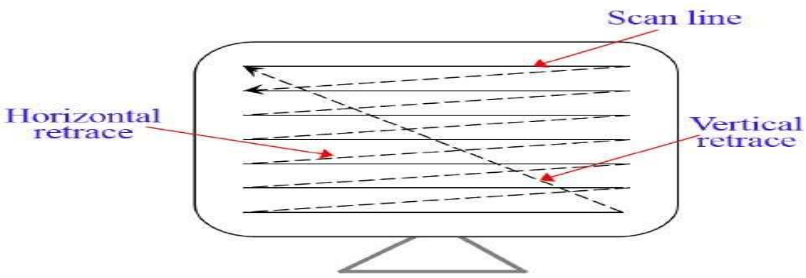
Q. Write a short note on Random scan display.

There are two different ways of scanning video displays. These are called as raster and random scans. In raster scan the electron beam follows a fixed path as shown in fig 1.7.1. The electron beam starts at top left corner of the screen and moves horizontally to the right . This defines a scan line. During the scan the intensity of the beam is modulated according to the 33pattern of the desired image along the line. At the right corner of the screen, the beam becomes off and moved back to the left edge of the screen at the starting point on next line.

This is shown as dotted line, which is called as horizontal retrace. This way of scanning is continued till the bottom right corner is reached. At this point, one scan is completed. The beam is then repositioned at the top left corner of the screen for starting another scan. This movement of beam from bottom right corner to top left corner is called as vertical retrace.

Display devices with random scan operate by directing the electron beam to only those parts of the screen where the picture is to be drawn. This is also called as vector graphics. This vector graphics depends on the ability of hardware to generate line vectors. Creation of diagrams using vector graphics becomes easier, so can be used in engineering and scientific drawings, whereas the raster graphics can be used in animation.

In random scan pen plotters and direct storage view tubes (DVST) devices are used, whereas in raster scan cathode ray tubes (CRT) are used. Since the cost of devices used for random scan is much higher than the cost of devices for raster scan, people are using raster scan devices heavily.



Write a note on Random scan display.

- Random scan display uses the technology to draw directly the pictures on the screen i.e. the beam is display on the positions where the picture has to be displayed.
- Random scan monitors draw a picture one line at a time and for this reasons are also referred to as vector display.
- Refresh rate on a random scan system depends on the number of lines to be displayed. Picture definition is stored as set of line drawing commands in an area of memory referred to as the **refresh display file.**

- Refresh display file is also called as the display list, display program, or refresh buffer.
- To display a specified picture, the system cycle through the set of commands in the display file, drawing each components line in turn.
- After all line drawing commands have been processed the system cycle back to the first line command in the list.
- Random-scan displays are designed to draw all the component lines of a picture 30 to 60 Times each second.

Write down the advantages and disadvantages of random scan display.

Advantages of random scan display are as follows:

- It has very high resolution and limited only by the monitor.
- In random scan, easy animations are possible.
- It requires loss memory.

Disadvantages of random scan display are as follows:

- It requires processor controlled beam.
- It cannot draw a complex image as it has limited density.
- It is very expensive in terms of color.

For random scan the host system generates a display file of graphics commands which is executed by the display processor. The display processor then just the frame buffer and turns the electron beams on/off.

Describe the difference between raster scan and random scan with respect to-

- How it stores information in memory?

-How it display stored information?

	Raster Scan	Random Scan
1) How it stores information in memory?	It stores information in memory area called the refresh buffer or frame buffer.	It stores information in memory area called the refresh display file.
2) How it displays stores information?	To display information the electron beam returns to the left side of the screen to being displaying the next scan line. At the end of each frame, the electron beam returns to the top left corner of the screen to being the next frame.	To display a specific picture, the system cycles through the set of commands in the display file, drawing each component line in turn. After all line drawing commands have been proceed, the system cycles back to the first line command in the list.

Differentiate between Random scan and Raster Scan Display.

Random scan	Raster scan
1) In random scan display, the beam is moved between the end points of the graphics primitives.	1) IN raster scan display, the beam is moved all over the screen one scan line at a time, From top to bottom and then back to top
2) Random display flickers when the number of primitives in the buffer becomes too large.	2) In raster display, the refresh process is independent of the complexity of the image.
3) Scan conversion is not required.	4) Scan conversion is required.
5) Scan conversion hardware is not required.	5) Scan conversion hardware is required.
6) Random display draw a continuous and smooth lines.	5) Raster display can display mathematically smooth lines, polygons, and boundaries of curved primitives only by approximating them with pixels on the raster grid.
7) Cost is more	6)Cost is low
8) Random display only draws lines and characters.	7) Raster display has ability to display areas filled with solid colors

2. What do you mean by graphics output primitives?

For creating pictures in graphics applications, we need a software package, which consists of library of functions that we can directly use within a programming language. This package is known as computer-graphics application programming interface (CG API). Since a graphics application consists of several component parts, the first thing we need to do is to describe the component parts of a scene. These components can be trees and terrain, atoms and molecules, or stars and galaxies, and soon. Then, for each type of scene, one needs to describe the structure of the individual objects with their coordinates within the scene Thus, in a graphics package, the functions which are used to describe the various picture components are known as graphics output primitives or simply primitives. The output primitives describing the geometry of objects are known as geometric primitives. Some examples of geometric primitives are point positions, straight line segments, circles, andsoon.

2.1 POINTS AND LINES

1 Point plotting is accomplished by converting a single coordinate position furnished by an application program into appropriate operations for the output device in use.

The formula for the point to be plotted is putpixel(x,y,color)

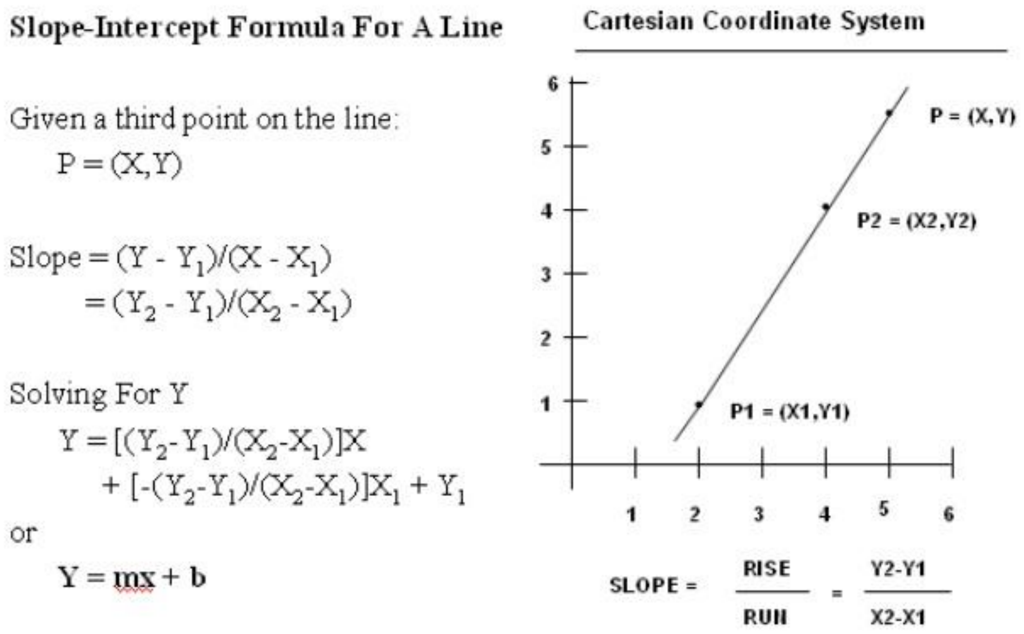
2.1 Give the procedure how line will be generated in analog devices (or vector-scan systems)

(Rastarization)

LINE-DRAWING ALGORITHMS

Design of Line and Circle Algorithms

Basic Math Review



3. Line drawing is accomplished by calculating intermediate positions along the line path between specified end points.

Precise definition of line drawing

4. Given two points P and Q in the plane, both with integer coordinates, determine which pixels on a raster screen should be on in order to make a picture of a unit-width line segment starting from P and ending at Q.

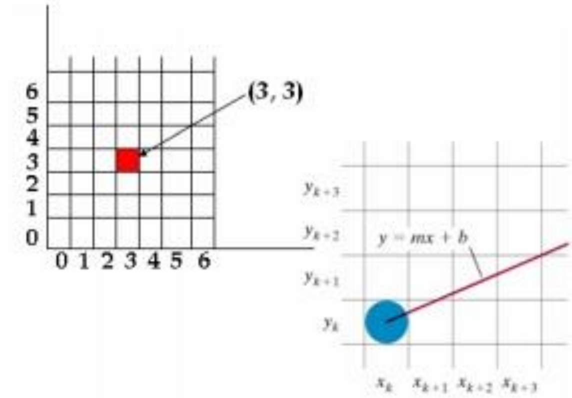


Figure 3-10

A section of the screen showing a pixel in column x_k on scan line y_k that is to be plotted along the path of a line segment with slope $0 < m < 1$.

The thinnest line is of one-pixel wide. We will concentrate on drawing a line of 1 pixel resolution. The Cartesian slope-intercept equation for a straight line is

$$y= m. x + b$$

m is the slope of the line and b is the y intercept.

Given the endpoints of a line segment. m = y2-y1 / x2-x1

$$b= y1-m.x1$$

5. Also for any given interval Δx along a line, we can compute the corresponding y interval Δy from Δy= m. x

6. Similarly we can obtain the x interval Δx corresponding to a specified Δy as Δx= Δy / m

7. These equations form the basis for determining deflection voltages in analog devices.

8. Also , for any given x interval Δx along a line, we can compute the corresponding y interval Δy from Δy= m. Δ x
9. These equations form the basis for determining deflection voltages in analog devices.
10. On Raster systems, lines are plotted with pixels, and step sizes in the horizontal and vertical directions are constrained by pixel separations. Hence we ought to “sample”

a line at discrete positions and determine the nearest pixel to the line at each sampled position

2.3 DDA Algorithm

Step 1 − Get the input of two end points (X0,Y0)(X0,Y0) and (X1,Y1)(X1,Y1).

Step 2 − Calculate the difference between two end points.

$dx = X_1 - X_0$

$dy = Y_1 - Y_0$

Step 3 − Based on the calculated difference identify put pixel setps.

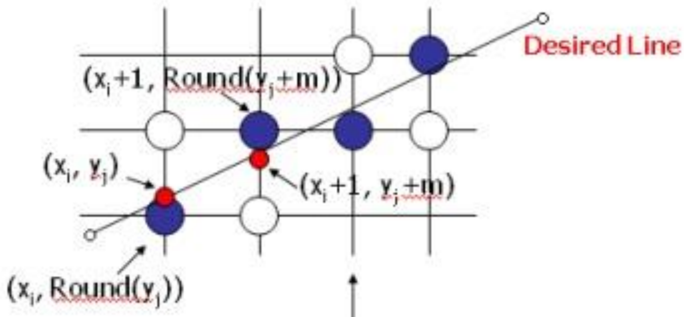
```
if(absolute(dx)> absolute(dy))

Steps= absolute(dx);

else

Steps= absolute(dy);
```

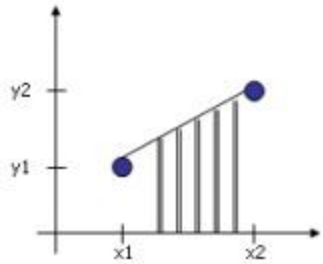
Step 4 − Calculate the increment in x coordinate and y coordinate.

```
Xincrement = dx / (float) steps;

Yincrement = dy / (float) steps;
```

Step 5 − Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing of the line

```
for(int v=0; v <Steps; v++)

{

  x = x +Xincrement;

  y = y +Yincrement;

  putpixel(Round(x),Round(y));

}
```





2.3.1Give the procedure for DDA algorithm.

Ans. The procedure for DDA algorithm is as follows:

Procedure line DDA (x0, y0, xend, yend : interger);l

Var

dx, dy, steps, k : interger;

xIncrement, yIncrement, x, y : real;

being

dx :=xend – x0;

dy :=yend – y0;

if abs (dx) > abs (dy), then step := abs (dx)

else steps := abs (dy);

xIncrement :=dx / steps;

yIncrement :=dy / steps;

x :=x0;

y :=y0;

setPixel (round (x), round(y),1);

for k :=1 to steps do

begin

x :=x+ xIncrement;

y :=y+ yIncrement;

setPixel (round (x), round (y),1)

end

end; (lineDDA)

The above procedure accepts the two integer endpoint pixel positions (x0,y0) and xend, yend) as an input. The horizontal and vertical differences between the endpoint positions are assigned to parameter dx and dy, resp. The difference with greater magnitude determines the value of parameter steps. Starting with pixel position (x0,y0) the offset needed at each step is determined so parameter the next pixel position along the line path and, thus, process is repeated step s times. If the magnitude of dx is greater than the magnitude of dy and x0 is less than xend, then the values of the increment sin the x and y directions are 1 and m, respectively. If the greater change is in the x-direction, but x0 is greater then xend , then the decrements -1 and –m are used to generate each new point on the line. Otherwise, there is a unit increment (or decrement) in the y-direction and an x increment (or decrement) of 1/m is used. It is assumed that points are to be plotted on bi-level intensity system.such that there is a cell to setpixel with an intensity value of 1.

Advantages and Disadvantages

Some of the advantages of the DDA algorithm are as follows:

- This algorithm is a faster method for calculating pixel position than the direct use of the line equation.
  y = mx + b
- It eliminates the floating-point multiplication which needs to be calculated in y = mx + b equation. This is possible because DDA algorithm makes use of raster characteristics and appropriate increments are applied in the x or y directions to move from one pixel position to another along the line path.

Some of the disadvantages of the DDA algorithm are as follows:

- A floating point addition is still needed to be done for determining each successive point.
- The cumulative error due to limited precision in the floating point representation may cause calculated points to frift away from their true position for long line segments.
- The rounding off operations and floating point arithmetic are time consuming.

Difference Between DDA Line Drawing Algorithm and Bresenhams Line Drawing Algorithm | DDA Algorithm vs. Bresenhams Algorithm

	Digital Differential Algorithm	**Bresenhams algorithm**
Arithmetic	DDA algorithm uses **floating points** i.e. **Real Arithmetic**.	Bresenhams algorithm uses **fixed points** i.e. **Integer Arithmetic**.
Operations	DDA algorithm uses **multiplication** and **division** in its operations.	Bresenhams algorithm uses only **subtraction** and **addition** in its operations.
Speed	DDA algorithm is rather **slowly** than Bresenhams algorithm in line drawing because it uses real arithmetic (floating-point operations).	Bresenhams algorithm is faster than DDA algorithm in line drawing because it performs only addition and subtraction in its calculation and uses only integer arithmetic so it runs significantly **faster**.
Accuracy & Efficiency	DDA algorithm is not as accurate and efficient as Bresenham algorithm.	Bresenhams algorithm is more efficient and much accurate than DDA algorithm.
Drawing	DDA algorithm can draw circles and curves but that are not as accurate as Bresenhams algorithm.	Bresenhams algorithm can draw circles and curves with much more accuracy than DDA algorithm.
Round Off	DDA algorithm round off the coordinates to integer that is nearest to the line.	Bresenhams algorithm does not **round off** but takes the incremental value in its operation.
Expensive	DDA algorithm uses an enormous number of floating-point multiplications so it is expensive.	Bresenhams algorithm is less expensive than DDA algorithm as it uses only addition and subtraction.

2.3.2 Implement the DDA algorithm to draw a line from (0,0) to (6,6).

Ans: Compute initial values:

(x0 ,y0) = (0,0)

(xend , yend) =(6,6)

dx = xend - x0 = 6-0 =6

dy = yend - y0 = 6-0= 6

steps =6

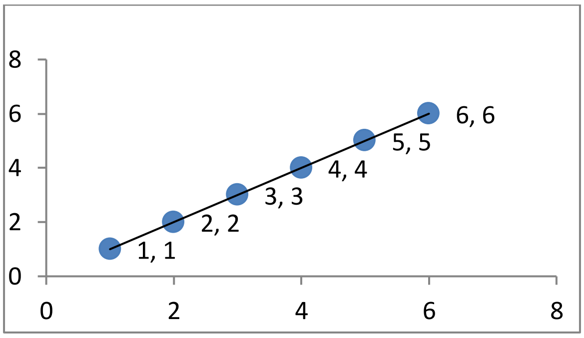
=dx/steps =6/6 =1

=dx/steps =6/6 =1

The following table indicates the value computed by algorithm

S.No	x	y
1	0	0
2	1	1
3	2	2
4	3	3
5	4	4
6	5	5
7	6	6

The pixels position along this line path are shown in Figure



2.4Give the Bresenham's line-drawing algorithm for a line with positive slope less than 1

The Bresenham's line drawing algorithm for |m| < 1 is as follows:

- Input the two line endpoint and store the left endpoint in (x_0, y_0)
- Load (x_0, y_0) into the frame buffer; that is, plot the first point.
- Calculate the value $\Delta x, \Delta y, 2\Delta y\ and\ 2\Delta y - 2\Delta x$,and obtain the starting value for the decision parameter as

$$d_0 = 2\Delta y - \Delta x$$

- At each x_i along the line, staritng at i = 0, perform the following test:

If ,$d_i < 0$, then the next point to plot is $(x_i + 1, y_1)$ and $d_{i+1} = d_i + 2\Delta y$

Otherwise, the next point to plot is $(x_i + 1, y_i + 1)$ and $d_{i+1} = d_i + 2\Delta y - 2\Delta x$

- Repeat step 4 $\Delta x - 1$times.

M<1		M>1	
P<0	**P>0**	**P<0**	**P>0**
P=P+2DY	**P=P+2DY-2DX**	**P=P+2DX**	**P=P+2DX-2DY**
X=X+1	**X=X+1**	**Y=Y+1**	**X=X+1**
Y=Y	**Y=Y+1**	**X=X**	**Y=Y+1**

2.4.1 Give the procedure for Bresenham's line-drawing algorithm.

Ans: for slopes in the range 0 < m <1 is as follows.

Procedure lineBres (xa, ya ,xb ,yb : interger);

Var

dx, dy, x, y, xEnd, d : interger;

begin

dx :=abs(xa - xb);

dy :=abs(ya - yb);

d := 2* dy - dx;

{determine which point to use as start, which as end}

If xa > xb then

Begin

x := xb;

y :=yb;

xEnd := xa;

end

else

begin

x :=xa;

y :=ya:

```
xEnd := xb;
end;
setPixel(x,y,1);
while x < xEnd do
begin
x := x+1;
if d <0 then d := d+2 * dy
else
begin
y := y+1;
d := d+2*(dy-dx);
end;
setPixel (x,y,1)
end;
end; {end of procedure}
```

In the above procedure, endpoint pixel positions for the line are passed to the procedure, and pixels are plotted from the left endpoint to the right endpoint. The cell to setPixel loads the intensity value 1 into the frame buffer at the specified (x,y) pixel position.

Advantages of Bresenham's line drawing algorithm.

- It is a faster and highly efficient incremental method for scan-converting lines than DDA algorithm.
- It uses only integer calculations such as addition/subtraction and bit shifting operations to give accurate results.
- The constants 2 are computed only once for each line which is to be scan-converted.
- This algorithm can be adapted to display circles and other curves.

The main drawback of such algorithm is that it is meant for basic line drawing, that is, the line displayed will have a jagged or stair-step appearance. To draw smooth lines, one has to look for a different algorithm.

2.4.2 Solve using Bresenham's line-drawing algorithm (20,10) to (30, 18).

Ans: Here, () = (20,10) and () = (30,18)

Plot the initial point () = (20,10) and determine the successive pixel positions along the line path from the decision parameter as follows:

For i = 0,

Here, > 0, as per the Bresenham's line-drawing algorithm, coordinates of next pixel positions () becomes (), that is (21,11)

And,

For i = 1,

Since, > 0, hence coordinates of next pixel position becomes (22,12)

And,

For i = 2,

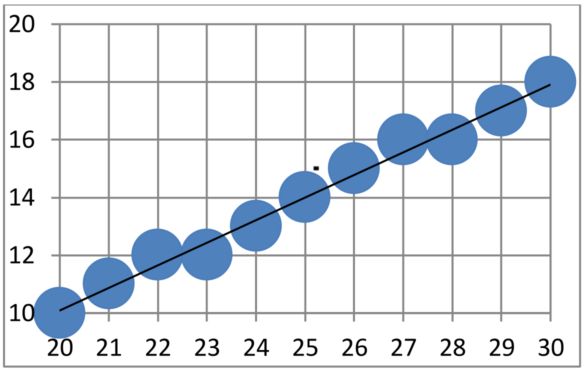
Since, > 0, hence coordinates of next pixel position becomes (23,12)

And,

Similarly, all the values for the entire successive pixel positions are shown in the table as follows:

0	6	(21,11)
1	2	(22,12)
2	-2	(23,12)
3	14	(24,13)
4	10	(25,14)
5	6	(26,15)
6	2	(27,16)
7	-2	(28,16)
8	14	(29,17)
9	10	(30,18)

The plot of the pixel generated along this line path is shown in figure.



2.5 CIRCLE/ CURVE -GENERATING ALGORITHMS

2.5.1 Properties of a circle:

- A circle is defined as a set of points that are all the given distance *(xc,yc)*. This distance relationship is expressed by the Pythagoras theorem in Cartesian coordinates as

$$(x – xc)2 + (y – yc)\ 2 = r2$$

- We could use this equation to calculate the points on the circle circumference by stepping along x-axis in unit steps from *xc-r* to *xc+r* and calculate the corresponding y values at each position as

$$y = yc +(-\ )\ (r2 – (xc –x\ )2)1/2$$

- This is not the best method:

- *Considerable amount of computation*

- *Spacing between plotted pixels is not uniform*

Polar co-ordinates for a circle

- We could use polar coordinates r and θ,

$$x = xc + r\cos\theta \quad y = yc + r\sin\theta$$

- A fixed angular step size can be used to plot equally spaced points along the circumference

■A step size of 1/r can be used to set pixel positions to approximately 1 unit apart for a continuous boundary But, note that circle sections in adjacent octants within one quadrant are symmetric with respect to the 45 deg line dividing the to octants

■Thus we can generate all pixel positions around a circle by calculating just the points within the sector from x=0 to x=y

■This method is still computationally expensive

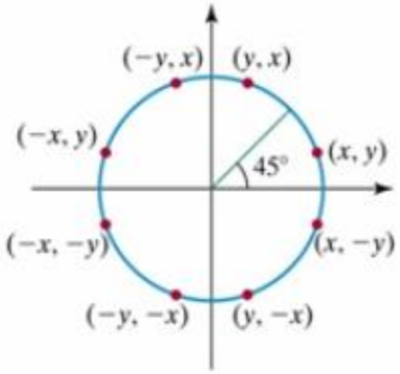


Figure 3-18

Symmetry of a circle. Calculation of a circle point (x, y) in one octant yields the circle points shown for the other seven octants.

2.5.2 Bresenham's Circle Drawing Algorithm:

1. Input radius 'r' of the circle.
2. Calculate the initial value of the decision parameter/variable
   $p_k = 3 - 2r$
3. Load the initial point of the circle in the frame buffer (0,r).
4. If we are using eight-way symmetry to plot the pixels, then until (x<=y) we have to perform following steps:
   if (p_0<0) then
   p = p + 4x + 6
   x = x + 1
   else
   p = p + 4(x-y) + 10
   x = x+1
   y =y-1
5. Plot(x,y).
6. Determine and plot the corresponding symmetry points in other octants also.
7. Stop.

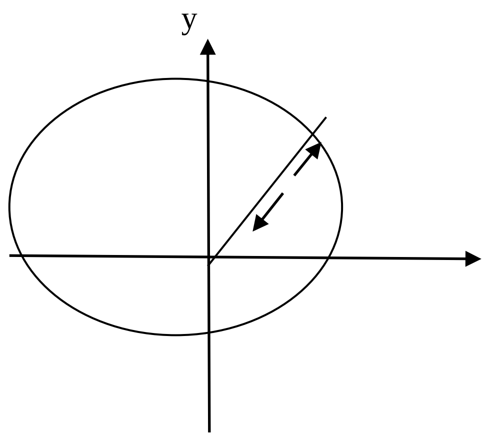
2.5.2.1 Example: Plot a circle by Bresenham's algorithm with whose radius is 3 and center is (0,0).

Solution: Initial point = (0,r) = (0,3)

Now, decision parameter at initial point p = 3 – 2r

= 3 – 2 * 3 = 3 – 6

= -3



Circle with r=3 and center at origin

Now p < 0 then S is chosen and x is incremented by one unit.

x = x + 1 = 1

y = y = 3

Next point to plot(1,3)

Now p = p + 4x + 6

= -3 + 4(1) + 6 = -3 + 4 + 6

= 7

Now p > 0 then p is chosen and x is incremented by one unit and y is decremented by one unit.

x = x + 1 = 1 + 1 = 2

y = 3 – 1 = 2

Next point to plot (2,2)

Since the value of x is equal to value of y, so no more further calculations are required in the octant from 90^0 to 45^0.

2.5.3 Midpoint Circle Algorithm:

1. Input radius r and circle center (x_c , y_c), and obtain the first point on the circumference of a circle entered on the origin as

   $(x_0, y_0) = (0,r)$
2. Calculate the initial value of the decision parameter as
   $p_0 = 5/4 – r$ where r is non-integer
   {for r an integer we take $p_0 = 1-r$}
3. At each x_k position, starting at k=0, perform the following test:
   If, $p_k<0$ the nextpoint along the circle centered on (0,0) is (x_{k+1}, y_k) and
   $P_{k+1} = p_k + 2x_{k+1} + 1$
   Otherwise, the next point along the circle is $(x_k + 1, y_k – 1)$ (i.e., $p_k>=0$)
   $P_{k+1} = p_k + 2(x_k+1) + 1-2y_{k+1}$
   Where $2x_{k+1} = 2x_k + 2$ and $2y_{k+1} = 2y_k – 2$
4. Determine symmetry points in the other seven octants.
5. Move each calculated pixel position (x,y) onto the circular path centered on (x_c , y_c) and plot the co0rdinat values.
   $x = x + x_c$; $y = y + y_c$
6. Repeat steps 3 through 5 until x>=y

1. Mid Point Circle Drawing Algorithm
2. Bresenham's Circle Drawing Algorithm

In this article, we will discuss about Mid Point Circle Drawing Algorithm.

Mid Point Circle Drawing Algorithm-

The points for other octacts are generated using the eight symmetry property.

1.1.1 Procedure-

Given-

- Centre point of Circle = (X_0, Y_0)
- Radius of Circle = R

The points generation using Mid Point Circle Drawing Algorithm involves the following steps-

1.1.2 Step-01:

Assign the starting point coordinates (X_0, Y_0) as-

- $X_0 = 0$
- $Y_0 = R$

1.1.3 Step-02:

Calculate the value of initial decision parameter P_0 as-

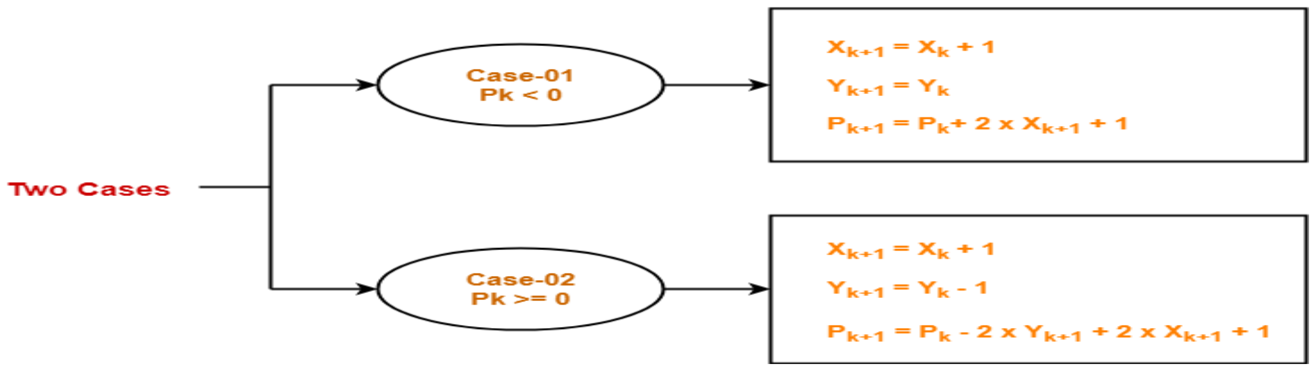
$$P_0 = 1 – R$$

1.1.4 step-03:

Suppose the current point is (X_k, Y_k) and the next point is (X_{k+1}, Y_{k+1}).

Find the next point of the first octant depending on the value of decision parameter P_k.

Follow the below two cases-



1.1.5 tep-04:

If the given centre point (X_0, Y_0) is not (0, 0), then do the following and plot the point-

- $X_{plot} = X_c + X_0$
- $Y_{plot} = Y_c + Y_0$

Here, (X_c, Y_c) denotes the current value of X and Y coordinates.

1.1.6 Step-05:

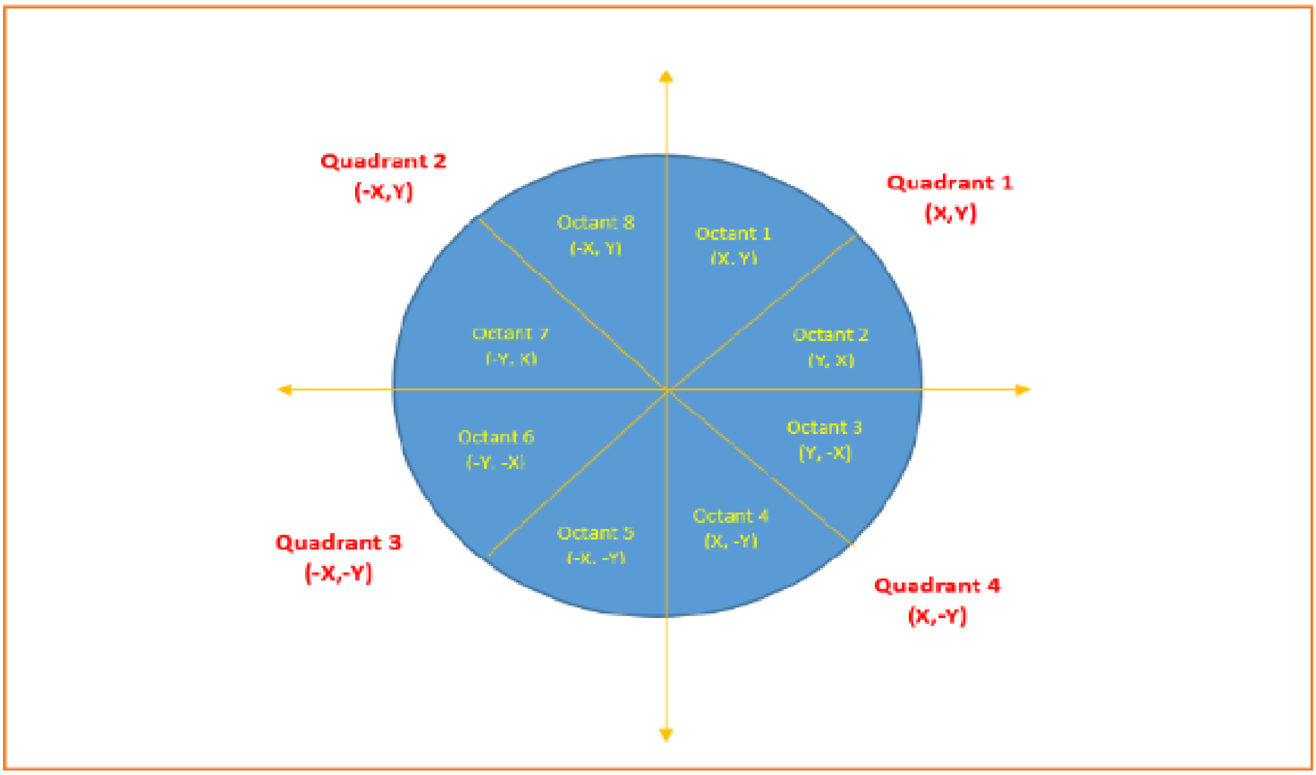
Keep repeating Step-03 and Step-04 until $X_{plot} >= Y_{plot}$.

1.1.7 Step-06:

Step-05 generates all the points for one octant.

To find the points for other seven octants, follow the eight symmetry property of circle.

This is depicted by the following figure-



PRACTICE PROBLEMS BASED ON MID POINT CIRCLE DRAWING ALGORITHM-

Problem-01:

Given the centre point coordinates (0, 0) and radius as 10, generate all the points to form a circle.

Solution-

Given-

- Centre Coordinates of Circle (X_0, Y_0) = (0, 0)
- Radius of Circle = 10

1.1.8 Step-01:

Assign the starting point coordinates (X_0, Y_0) as-

- $X_0 = 0$
- $Y_0 = R = 10$

1.1.9 Step-02:

Calculate the value of initial decision parameter P_0 as-

$P_0 = 1 – R$

$P_0 = 1 – 10$

$P_0 = -9$

1.1.10 Step-03:

As $P_{initial} < 0$, so case-01 is satisfied.

Thus,

- $X_{k+1} = X_k + 1 = 0 + 1 = 1$
- $Y_{k+1} = Y_k = 10$
- $P_{k+1} = P_k + 2 \times X_{k+1} + 1 = -9 + (2 \times 1) + 1 = -6$

1.1.11 Step-04:

This step is not applicable here as the given centre point coordinates is (0, 0).

1.1.12 tep-05:

Step-03 is executed similarly until $X_{k+1} >= Y_{k+1}$ as follows-

P_k	P_{k+1}	(X_{k+1}, Y_{k+1})
		(0, 10)
-9	-6	(1, 10)
-6	-1	(2, 10)
-1	6	(3, 10)
6	-3	(4, 9)
-3	8	(5, 9)
8	5	(6, 8)
		(7,7)
Algorithm Terminates **These are all points for Octant-1.**		

Algorithm calculates all the points of octant-1 and terminates.

Now, the points of octant-2 are obtained using the mirror effect by swapping X and Y coordinates.

Octant-1 Points	Octant-2 Points
(0, 10)	**(7,7)**
(1, 10)	**(8, 6)**
(2, 10)	**(9, 5)**
(3, 10)	**(9, 4)**
(4, 9)	**(10, 3)**
(5, 9)	**(10, 2)**
(6, 8)	**(10, 1)**
(7,7)	**(10, 0)**
These are all points for Quadrant-1.	

Now, the points for rest of the part are generated by following the signs of other quadrants.

The other points can also be generated by calculating each octant separately.

Here, all the points have been generated with respect to quadrant-1-

Quadrant-1 (X,Y)	Quadrant-2 (-X,Y)	Quadrant-3 (-X,-Y)	Quadrant-4 (X,-Y)
(0, 10)	(0, 10)	(0, -10)	(0, -10)
(1, 10)	(-1, 10)	(-1, -10)	(1, -10)
(2, 10)	(-2, 10)	(-2, -10)	(2, -10)
(3, 10)	(-3, 10)	(-3, -10)	(3, -10)
(4, 9)	(-4, 9)	(-4, -9)	(4, -9)
(5, 9)	(-5, 9)	(-5, -9)	(5, -9)
(6, 8)	(-6, 8)	(-6, -8)	(6, -8)
(7,7)	(-7,7)	(-7,-7)	(7,-7)
(8, 6)	(-8, 6)	(-8, -6)	(8, -6)
(9, 5)	(-9, 5)	(-9, -5)	(9, -5)
(9, 4)	(-9, 4)	(-9, -4)	(9, -4)
(10, 3)	(-10, 3)	(-10, -3)	(10, -3)
(10, 2)	(-10, 2)	(-10, -2)	(10, -2)
(10, 1)	(-10, 1)	(-10, -1)	(10, -1)
(10, 0)	(-10, 0)	(-10, 0)	(10, 0)

These are all points of the Circle.

Figure :-

Problem-02:

Given the centre point coordinates (4, -4) and radius as 10, generate all the points to form a circle.

Solution-

Given-

- Centre Coordinates of Circle (X_0, Y_0) = (4, -4)
- Radius of Circle = 10

Example: An arc is to be drawn from (2,10) to (8,8) in a circle of radius 10 using Mid-point circle algorithm.

Solution:

3 ATTRIBUTES

3.1 LINE ATTRIBUTES

Basic attributes of a straight line segment are its type, its width, and its color. In some graphics packages, lines can also be displayed **using** selected pen or brush options.

Line Type

line-type attribute - solid lines, dashed lines, and dotted lines.
We modify a line drawing algorithm to generate such lines by setting the length and spacing of displayed solid sections along the line path.

A dashed line could be displayed by generating an inter dash spacing that is equal to the length of the solid sections. Both the length of the dashes and the inter dashs pacing are often specified as user options. **A** dotted line can be displayed by generating very short dashes with the spacing equal to or greater than the dash size. Similar methods are used to produce other line-type variations.

To set line type attributes in a **PHICS** application program, a user invokes the function

setLinetype **(It)**

where parameter I t is assigned a positive integer value of 1,2,3, or 4 to generate lines that are, respectively, solid, dashed, dotted, or dash-dotted.

Line Width

We set the line-width attribute with the command: Line-width parameter lr. is assigned a positive number to indicate the relative width of the line to be displayed. A value of *1* specifies a standard-width line. On.

For lines with slope magnitude greater than 1, we can plot thick lines with horizontal spans, alternately picking up pixels to the right and left of the line path.

Problem with implementing width options using horizontal or vertical pixel spans is that the method produces **lines** whose ends are horizontal or vertical regardless of the slope of the line. **This** effect is more noticeable with very thick lines. We can adjust the shape of the **line** ends to give them a better appearance by adding line caps

One kind of line cap is the butt cap obtained by adjusting the end positions of the component parallel **lines** so that the thick line is displayed with square ends that are perpendicular to the line path. If the specified line has slope m, the square end of the thick line has slope **- l / m .**

Another line cap is the round cap obtained by adding a filled semicircle to each but cap. The circular arcs are cantered on the line endpoints and have a diameter equal to the line thickness.

A third type of line cap is the projecting square cap. Here, we simply extend the line and add butt caps that are positioned one-half of the line width beyond the specified endpoints.

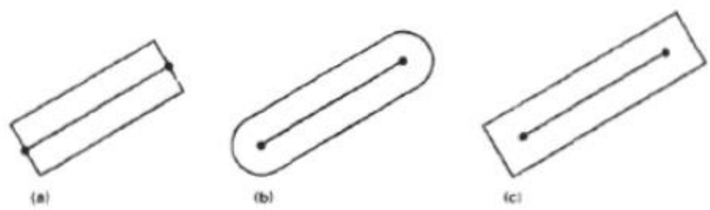


Figure 4-5
Thick lines drawn with (a) butt caps, (b) round caps, and (c) projecting square caps.

We can generate thick polylines that are smoothly joined at the cost of additional processing at the segment endpoints.

A miter join is accomplished by extending the outer boundaries of each of the two lines until they meet.
A round join is produced by capping the connection between the two segments with a circular boundary whose diameter is equal to the linewidth.
And a bevel join is generated by displaying the line segments with butt caps and filling in the triangular gap where the segments meet.

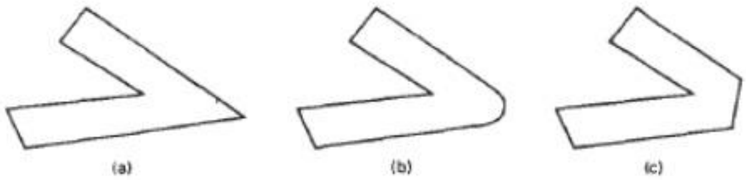


Figure 4-6
Thick line segments connected with (a) miter join, (b) round join, and (c) bevel join.

Pen and Brush Options
lines can be displayed with pen or brush selections. ***Op***tions in this category include shape, size, and pattern.

These shapes can be stored in a pixel mask that identifies the array of pixel positions that are to be set along the line path. Lines generated with pen (or brush) shapes canbe displayed in various widths by changing the size of the mask.

Line Color

When a system provides color (or intensity) options, a parameter giving the current

color index is included in the list of system-attribute values. A polyline routine displays a line in the current color by setting this color value in the frame buffer at pixel locations along the line path using the **setpixel** procedure.

The number of color choices depends on the number of bits available per pixel in the frame buffer.We set the line color value in**PHlCS** with the function

Set PolylineColourIndex (le)

3.2 CURVE ATTRIBUTES

Parameters for curve attributes are the **same** as those for line segments. We can display curves with varying colors, widths, dotdash patterns, and available pen or brush options.

AREA FILL ATTRIBUTES

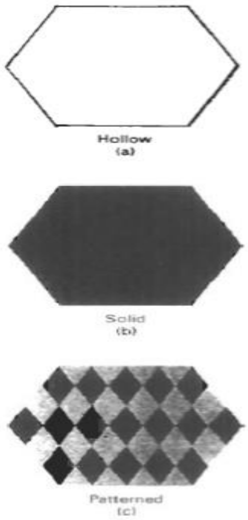


Figure 4-18

AREA-FILL ATTRIBUTES

Options for filling a defined region include a choice between a solid color or a patterned fill and choices for the particular colors and patterns. These fill options can be applied to polygon regions or to areas defined with curved boundaries, depending on the capabilities of the available package. In addition, areas can be painted using various brush styles, colors, and transparency parameters.

Fill Styles

Areas are displayed with three basic fill styles: hollow with a color border, filled with a solid color, or filled with a specified pattern or design. A basic fill style is selected in a PHIGS program with the function

```
setInteriorStyle (fs)
```

Values for the fill-style parameter `fs` include *hollow*, *solid*, and *pattern* (Fig. 4-18). Another value for fill style is *hatch*, which is used to fill an area with selected hatching patterns—parallel lines or crossed lines—as in Fig. 4-19. As with line attributes, a selected fill-style value is recorded in the list of system attributes and applied to fill the interiors of subsequently specified areas. Fill selections for parameter `fs` are normally applied to polygon areas, but they can also be implemented to fill regions with curved boundaries.

Hollow areas are displayed using only the boundary outline, with the interior color the same as the background color. A solid fill is displayed in a single color up to and including the borders of the region. The color for a solid interior or for a hollow area outline is chosen with

```
setInteriorColourIndex (fc)
```

where fill-color parameter `fc` is set to the desired color code. A polygon hollow fill is generated with a linedrawing routine as a closed polyline

Pattern Fill

We select fill patterns with

```
setInteriorStyleIndex (pi)
```

where pattern index parameter `pi` specifies a table position. For example, the following set of statements would fill the area defined in the `fillArea` command with the second pattern type stored in the pattern table:

```
setInteriorStyle (pattern);
setInteriorStyleIndex (2);
fillArea (n, points);
```

Separate tables are set up for hatch patterns. If we had selected *hatch* fill for the interior style in this program segment, then the value assigned to parameter `pi` is an index to the stored patterns in the hatch table.

For fill style *pattern*, table entries can be created on individual output devices with

```
setPatternRepresentation (ws, pi, nx, ny, cp)
```

TABLE 4-3
A WORKSTATION PATTERN TABLE WITH TWO ENTRIES, USING THE COLOR CODES OF TABLE 4-1

Index (*pi*)	*Pattern* (*cp*)
1	$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
2	$\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$

3.3 CHARACTER ATTRIBUTES

The appearance of displayed characters is controlled by attributes such as font, size, color, and orientation. Attributes can be set for entire character strings (text) and for individual characters defined as marker symbols.

3.4 TEXT ATTRIBUTES

There are a great many text options that can be made available to graphics programmers.

First of **all,** there is the choice of font (or typeface), which is a set of characters with a particular design style such as New York, Courier, Helvetica, London, 'Times Roman, and various special symbol groups.

The width only of text can be set with the function

```
setCharacterExpansionFactor (cw)
```

where the character-width parameter cw is set to a positive real value that scales the body width of characters. Text height is unaffected by this attribute setting. Examples of text displayed with different character expansions is given in Fig. 4-27.

Spacing between characters is controlled separately with

```
setCharacterSpacing (cs)
```

where the character-spacing parameter cs can be assigned any real value. The value assigned to cs determines the spacing between character bodies along print lines. Negative values for cs overlap character bodies; positive values insert space to spread out the displayed characters. Assigning the value 0 to cs causes text to be displayed with no space between character bodies. The amount of spacing to be applied is determined by multiplying the value of cs by the character height (distance between baseline and capline). In Fig. 4-28, a character string is displayed with three different settings for the character-spacing parameter.

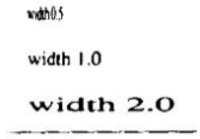


Figure 4-27
The effect of different character-width settings on displayed text.

Virtual reality is an artificial environment that is created with software and presented to the user in such a way that the user suspends belief and accepts it as a real environment. On a computer, virtual reality is primarily experienced through two of the five senses: sight and sound.