CST 304 COMPUTER GRAPHICS
&
IMAGE PROCESSING

Module – 1(Basics of Computer graphics and Algorithms) Basics of Computer Graphics and its applications. Video Display devices- Refresh Cathode Ray Tubes, Random Scan Displays and systems, Raster scan displays and systems. Line drawing algorithms- DDA, Bresenham's algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bresenham's algorithm.

"*A picture is worth a thousand words*"

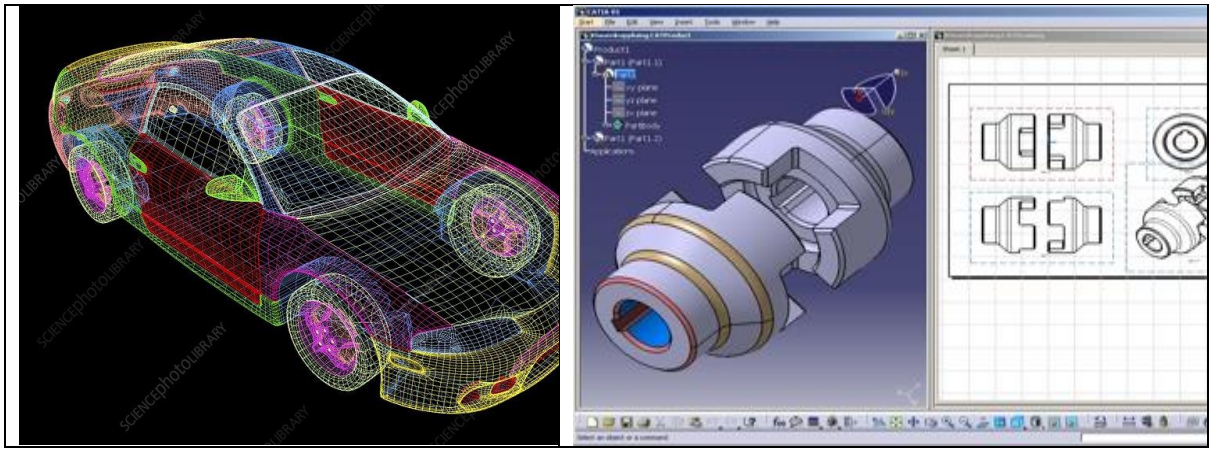
What is Computer Graphics?

Computer graphics generally refers to the specific creation, storage, and manipulation of image data using a digital computer with the help of a specialized graphical system.
Which studies methods for digitally synthesizing and manipulating visual content.

Applications of Computer Graphics

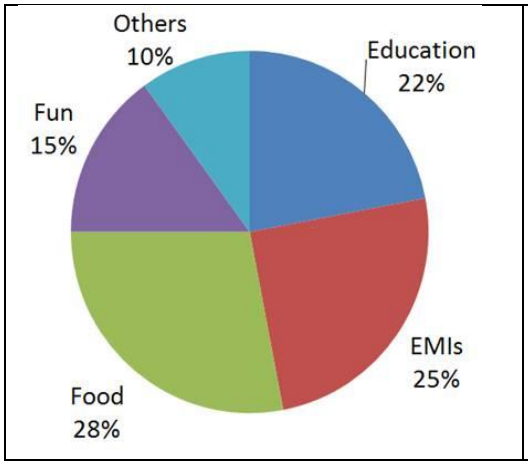
Computer-Aided Design

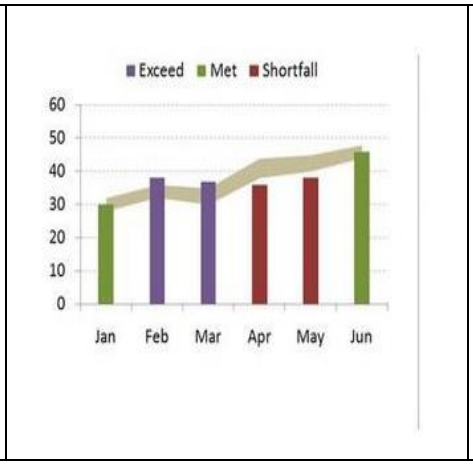
CAD methods are now routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft computers, textiles, and many other products.
For some design applications, objects are first displayed in a wireframe outline form that shows the overall shape and internal features of objects. Wireframe displays allow designers to quickly see the effects of interactive adjustments to design shapes.
Animations are often used in CAD applications. Real-time animations using wireframe displays on a video monitor are useful for testing the performance of a vehicle or system.

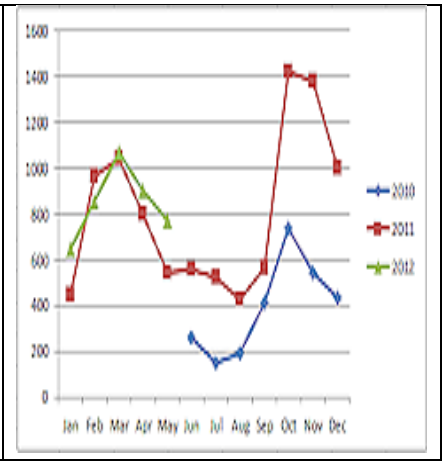


Presentation Graphics

Presentation Graphics programs are used to produce illustrations for reports or to generate PowerPoint presentations. It 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.





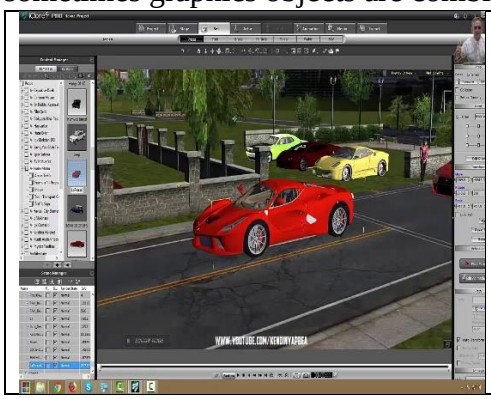


Computer Art

Computer graphics methods are widely used in both fine art and commercial art applications. Artist's paintbrush programs (Lumena), Pixel Paint and Super Paint, symbolic mathematics packages (Mathematica), CAD packages, desktop publishing software, animation packages that provide facilities for designing object shapes, and specifying object motions.
We have now Adobe illustrator, affinity designer, sketch, Corel Draw, gravit designer etc .

Entertainment

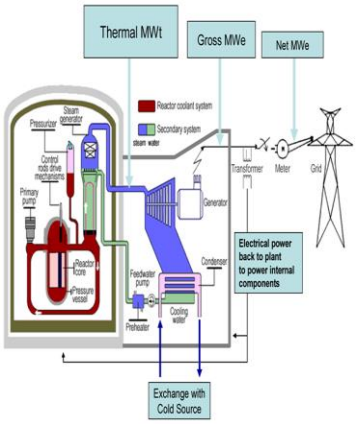
Computer graphics methods are now commonly used in making motion pictures, music videos, and television shows. Sometimes graphics objects are displayed by themselves, and sometimes graphics objects are combined with the actors and live scenes.



A common graphics method employed in many commercials is ***morphing***, where one object is transformed into another.

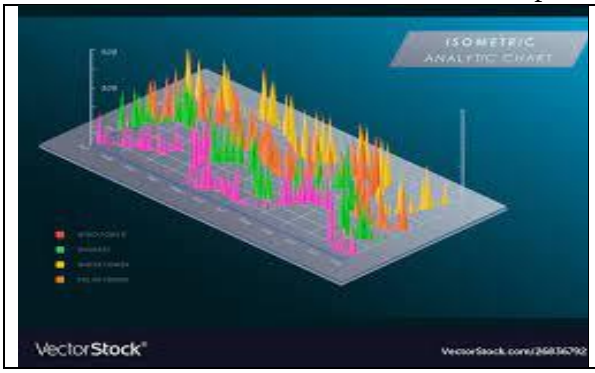
Education and Training

Computer-generated models of physical, financial, and economic systems are often used as educational aids. Models of physical systems, physiological systems, population trends, or equipment, such as color-coded diagrams can help trainees to understand the operation of the system.



Visualization

Scientific engineers, medical personnel, business analysts, and others often need to analyse large amounts of information or study the behaviour of certain processes. But if the data are converted to a visual form the trends and patterns are often immediately apparent.



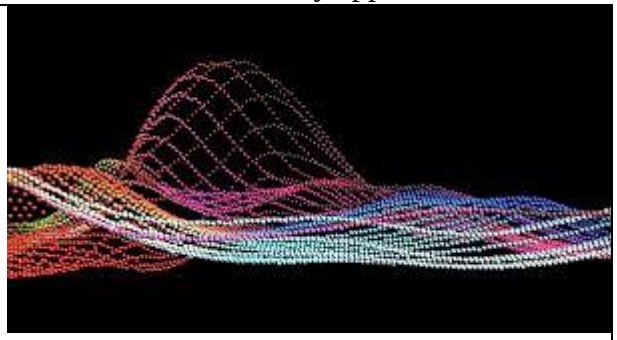
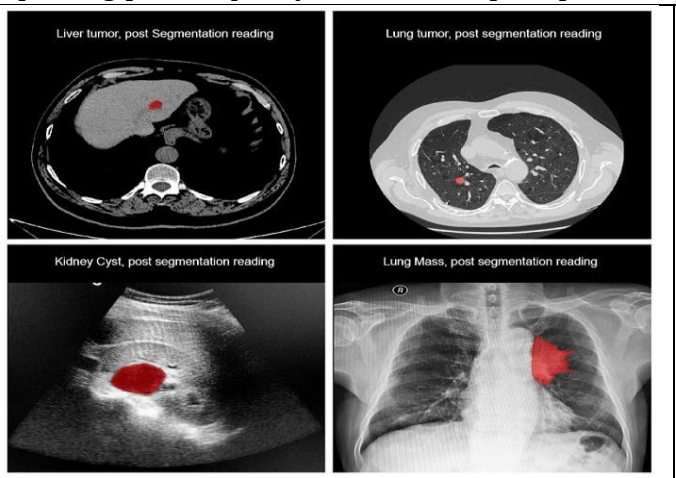


Image Processing

In computer graphics, a computer is used to create a picture.

Image processing, on the other hand, applies techniques to modify or interpret existing pictures, such as photographs and TV scans. Two principal applications of image processing are improving picture quality and machine perception of visual information as used in robotics.



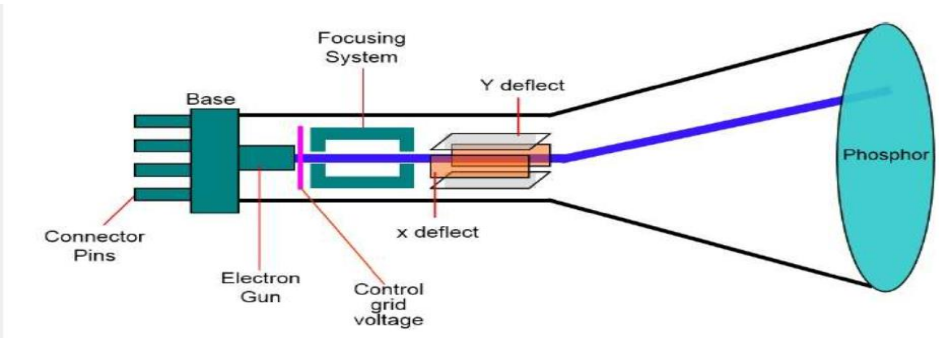
Video Display Devices

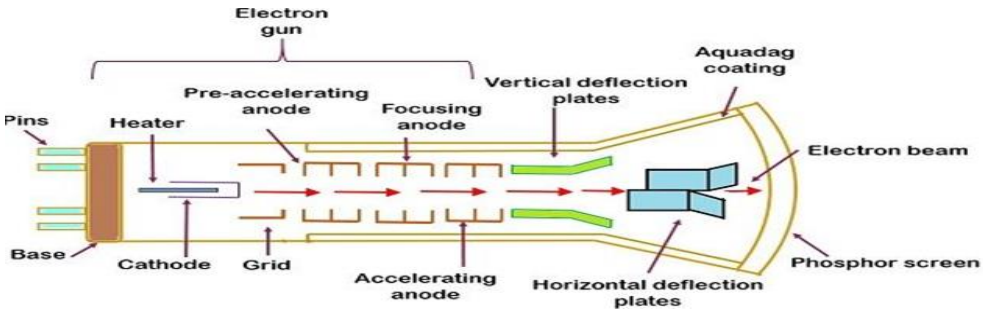
Refresh Cathode Ray Tubes

In a graphic system, the video monitor is a primary output device and the operation of most the video monitors is based on the Cathode Ray Tube.

In a cathode-ray tube a **beam of electrons**, emitted by an **electron gun**, passes through **focusing and deflection systems**, that direct the beam towards a specific position on the **phosphor-coated screen.** The part of the phosphor screen emits light where the electron beam strikes it. The light emitted by the phosphor fades very rapidly, one way to keep the phosphor glowing is to redraw the picture repeatedly by quickly directing the electron beam back over the same points. This type of display is called a **refresh CRT**.

Computer monitors often have a maximum **refresh rate.** This number, measured in hertz (Hz), determines how many times the screen is redrawn each second. Typical refresh rates for CRT monitors include 60, 75, and 85 Hz.





The primary components of an electron gun in a CRT are the heated metal cathode and a control grid. Heat is supplied to the cathode by directing a current through a coil of wire, called the filament, inside the cylindrical cathode structure. This causes electrons to be boiled off the hot cathode surface. In the vacuum inside the CRT envelope, the free negatively charged electrons are then accelerated towards the phosphor coating by a high positive voltage.

The amount of light emitted by the phosphor coating depends on the number of electrons striking the screen. We specify the intensity level for individual screen positions with graphics software commands.

Note:

A computer monitor's **resolution** refers to the approximate number of pixels the device is capable of displaying. It's expressed as the number of horizontal dots by the number of vertical dots; for example, an 800 x 600 resolution means the device can show 800 pixels across by 600 pixels down. In total, this screen displays 480,000 pixels.

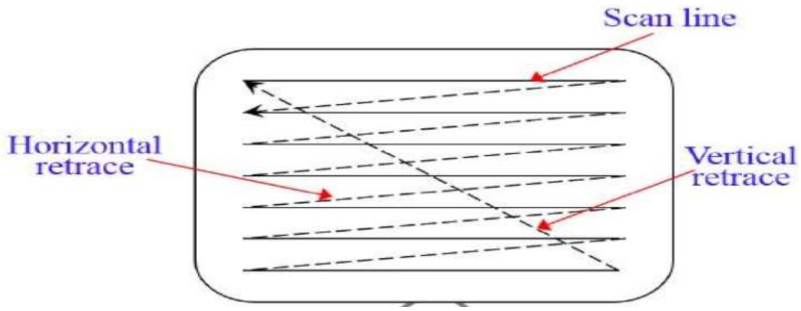
The **aspect ratio** is the ratio between the width and height of the screen. Eg: 32:9, 21:9, 16:9 etc.

Raster Scan Displays are the most common type of graphics monitor which employs CRT. It is based on television technology. In a 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 the frame buffer and painted on-screen taking one row at a time. Each screen point is referred to as a pixel or pel (shortened form of picture element).

The **refresh rate** (or "vertical refresh rate", "vertical scan rate") is the number of times per second that a raster-based display device redraws images.

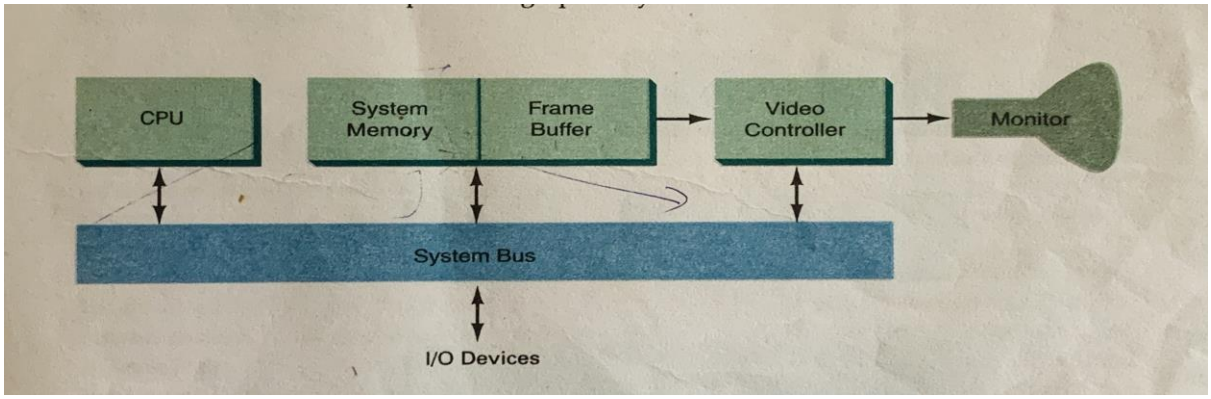
On cathode ray tube (CRT) displays, higher refresh rates produce less flickering, thereby reducing eye strain.

In raster scan systems refreshing is 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, the electron beam begins to display the next scan line after returning to the left side of the screen. The return to the left of the screen after the refresh of each scan line is known as the **horizontal retrace** of the electron beam. At the end of each frame, the electron beam returns to the top left corner and begins the next frame, this movement is known as **vertical retrace**.

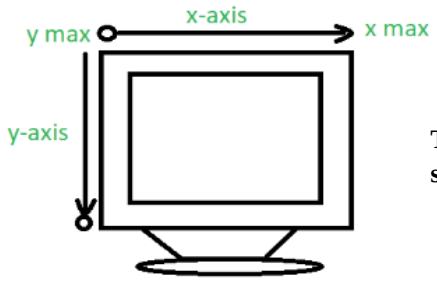


Raster Scan System

In addition to the central processing unit, a special purpose processor called the **video controller or video display controller** is used to control the operation of the display device .
A fixed area of the system memory is reserved for the frame buffer, and the video controller is given direct access to the frame buffer memory.
Frame buffer locations and corresponding screen positions are referenced in Cartesian coordinates.



Architecture of a simple raster graphics system



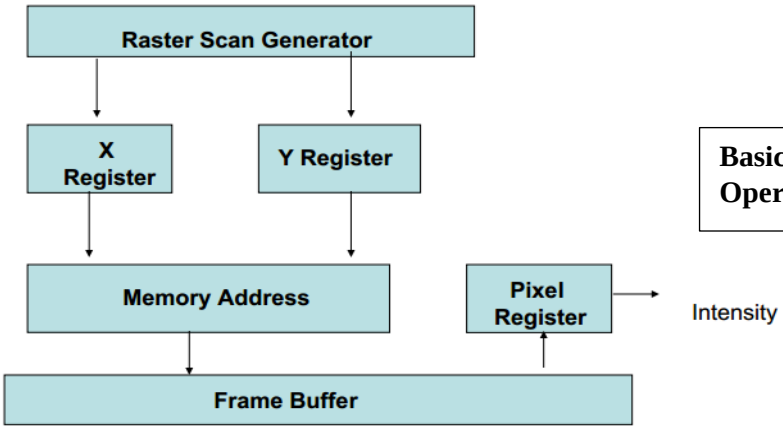
The origin of the coordinate system for identifying screen positions is usually specified in the lower-left corner

Basic video controller refresh operations

Note: Scanning start from top left corner, so coordinate for first pixel is (xmin,ymax) .

Two registers are used to store the coordinates of the screen pixels. Initially, the x registers are used to store the coordinates of the screen pixels. Initially x register is set to 0 and y register is set to ymax .The value stored in the frame buffer for this pixel position is then retrieved and used to set the intensity of the CRT beam . Then the x register is incremented by 1, and the process is repeated for the next pixel on the top scan line. This procedure is repeated for each

pixel along the scan line. After the last pixel on the top scan line has been processed, the x register is reset to 0 and the y register is decremented by 1. Pixels along this scan line are then processed in turn and the procedure is repeated for each successive scan line.
After cycling through all pixels along the bottom can line (y=0), the video controller resets the registers to the first-pixel position on the top scan line and the refresh process starts over.
Usually, cycle time is too slow if processing pixels one after another, to speed up pixel processing, video controllers can retrieve multiple pixel values from the refresh buffer on each pass. The multiple pixel intensities are then stored in a separate register and used to control the CRT beam intensity for a group of adjacent pixels. When that group of pixels has been processed, the next block of pixel values is retrieved from the frame buffer.



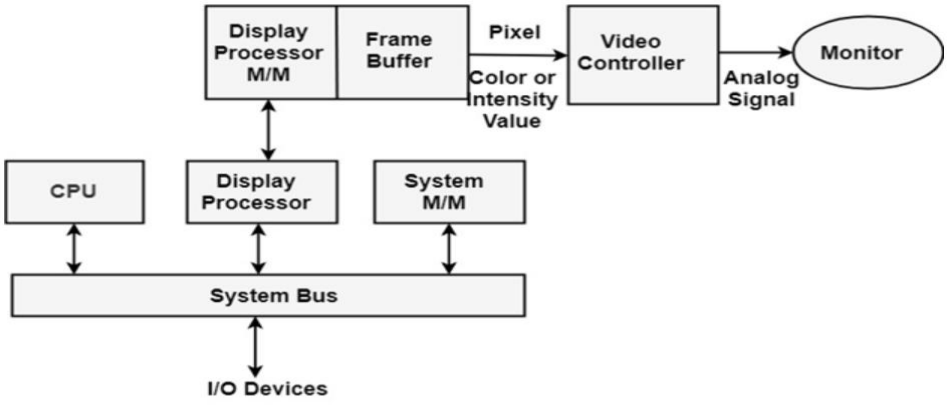
Basic Video Controller Refresh Operations

Note: In high-quality systems, for example, two frame buffers are often provided so that one buffer can be used for refreshing while the other is being filled with intensity values.

Raster -Scan Display Processor

Figure shows a raster system containing a separate display processor, referred to as a graphics controller or a display coprocessor. The purpose of the display processor is to free the CPU from the graphics chores. In addition to the system memory, a separate display processor memory area can also be provided.

A major task of the display processor is digitizing a picture definition given in an application program into a set of pixel intensity values for storage in the frame buffer. This digitization process is called scan conversion.



Architecture of a raster -graphics system with a display processor.

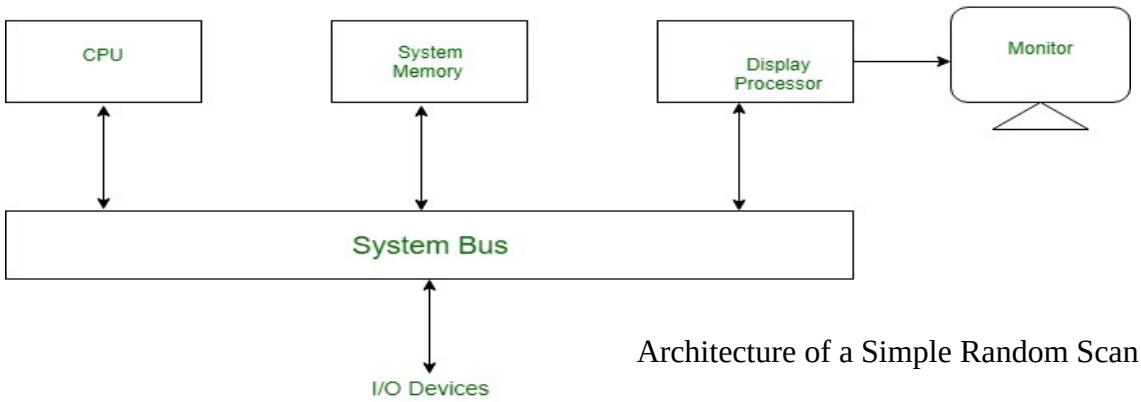
Random Scan Display

In **Random-Scan Display** electron beam is directed only to the areas of screen where a picture has to be drawn. It is also called **vector display**, as it draws picture one line at time. It can draw and refresh component lines of a picture in any specified sequence. Pen plotter is an example of random-scan displays.

The number of lines regulates refresh rate on random-scan displays. An area of memory called **refresh display files** stores picture definition as a set of line drawing commands. The system returns back to the first-line command in the list after all the drawing commands have been processed. High-quality vector systems can handle around 100, 00 short lines at this refresh rate. Faster refreshing can burn phosphor. To avoid this every refresh cycle is delayed to prevent a refresh rate greater than 60 frames per second.

Random Scan System

Input in the form of an application program is stored in the system memory along with graphics package. Graphics package translates the graphic commands in the application program into a display file stored in system memory. This display file is then accessed by the display processor to refresh the screen. The display processor cycles through each command in the display file program. Sometimes the display processor in a random scan is referred to as *Display Processing Unit / Graphics Controller*.



Architecture of a Simple Random Scan System

RASTER SCAN	RANDOM SCAN
While the resolution of raster scan is lesser or lower than random scan	The resolution of random scan is higher than raster scan
While the cost of raster scan is lesser than random scan.	It is costlier than raster scan.
It stores picture definition as a set of intensity values of the pixels in the frame buffer.	It stores picture definition as a set of line commands in the memory.
While in raster scan, any alteration is not so easy.	In a random scan, any alteration is easy in comparison of a raster scan.
The refresh rate is 60 to 80 frames per second and is independent of picture complexity.	The refresh rate depends on the number of lines to be displayed i.e. 30 to 60 times per second.
Eg: TV Sets	Eg : Pen Plotter

Color CRTMonitors

A CRT monitor displays colour pictures by using a combination of phosphors that emit different coloured light. By combining the emitted light from the different phosphors, a range of colours can be generated. The two basic techniques for producing colour displays with a CRT are the **beam-penetration method and the shadow-mask method.**

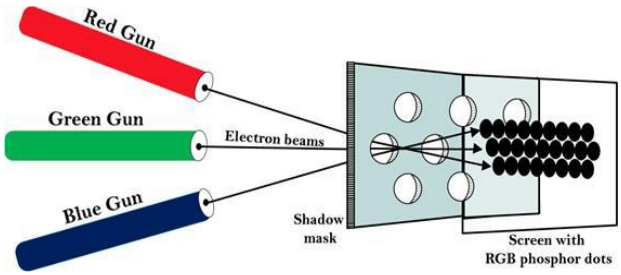
The Beam Penetration Method

It is used with random scan monitors. **Two layers of phosphor** , usually **red and green** , are coated on to the inside of the CRT screen, and the displayed color depends on how far the electron beam penetrates into the phosphor layers. A beam of slow electrons excites only the outer red layer. A beam of very fast electrons 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 colours, orange and yellow.The screen colour at any point, is controlled by the **beam acceleration voltage.**Beam penetration has been an **inexpensive** way to produce colour in random scan monitors. But only **four colors** are possible, and the **quality of pictures is not as good** as with other methods.

The Shadow-Mask Method

It is commonly used in raster scan systems (including color TV) because they produce a much wider range of colors than the beam penetration method.A 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.This type of CRT has **three electron guns**, one for each color dot, and **a shadow-mask grid** just behind the phosphor-coated screen.

Delta-Delta Shadow Mask Method



Commonly used in color CRT systems.
The three electron beams are deflected and focused as a group onto the shadow mask, which contains a series of holes aligned with the phosphor dot patterns. When the three beams pass through a hole in the shadow mask, they activate a **dot triangle**, which appears as a small color spot on the screen. The phosphor dots in the triangles are arranged so that each electron beam can activate only its corresponding color dot when it passes through the shadow mask.
We obtain color variations in a shadow mask CRT by varying the **intensity levels of the three electron beams.** By turning off the red and green guns, we get only the color coming from the blue phosphor.
Other combinations of beam intensities produce a small light spot for each pixel position since our eyes tend to merge the three colors into one composite.
The color we see depends on the amount of excitation of the red, green, and blue phosphors.

A **white area** is a result of activating all three dots with equal intensity.

Yellow is produced with the green and red dots only, **magenta** is produced with the blue and red dots, and **cyan** shows up when blue and green are activated equally.

Direct-View Storage Tubes

An alternative method for maintaining a screen image is to store the picture information inside the CRT instead of refreshing the screen.
A direct-view storage tube (DVST) stores the picture information as a **charge distribution just behind the phosphor-coated screen.**
Two electron guns are used in a DVST. One, the primary gun, is used to store the picture pattern; the Second, the flood gun, maintains the picture display.

Advantages over refresh CRT
•Very complex pictures can be displayed at very high resolutions without flicker.

Disadvantages of DVST systems

•They do not display colour.

•The erasing and redrawing process can take several seconds for a complex picture.

Problems

1)Suppose you have a raster system designed using an 8 inches x 10 inches screen with a resolution of 100 pixels per inch in each direction. What frame buffer size is required if 6 bits are stored per pixel in the buffer?

Answer : Resolution =8 inch x10 inch

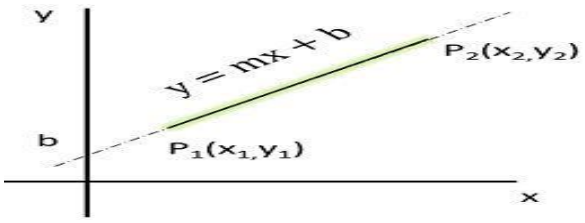
Each inch with 100 pixels so total 8x100 x 10x100 =800x1000 pixels .
One pixel can store 6 bits ,800x1000x6=4800000 bits =600000 bytes (1 byte=8 bits)
=**6×10^5 bytes .**

2)How much time is spent scanning across each row of pixels during screen refresh on a raster system with a resolution of 1280*1024 and are the fresh rate of 60 frames per second?

That means system contains 1024 scan lines and each scan line contains 1280 pixels.

refresh rate = 60 frame/sec. So, 1 frame takes = 1/60 sec.
Since resolution = 1280 X 1024 1 frame buffer consist of 1024 scan lines.
It means then 1024 scan lines takes 1/60 sec Therefore,
1 scan line takes , 1/(60 x1024)=**0.058 sec**

Line Drawing Algorithms



Scan converting a line

The slop Intercept equation for a straight line is **y=m.x +b** where m is slope of the line and b is the y-intercept.

Given that two endpoints of a line segment are specified at positions(x1,y1) and (x2,y2) then

m= y2-y1/x2-x1 ……..(eq1) ; b=y1-m.x1 ……..(eq2)

another representation → m=dx/dy where dy=y2-y1 and dx=x2-x1

dy=m.dx	**dx=dy/m**

Digital Differential Analyzer (DDA) Line Drawing Algorithm

DDA Algorithm is the simplest line drawing algorithm

Case 1: If slope (m) is less than or equal to 1, we sample at unit x intervals (**dx=1**) and

compute each successive y value as $y_{k+1} = y_k + m$ **……..(eq3)**

Case 2: If the slope (m) is greater than 1 , we sample at unit y intervals (**dy=1**) and compute each successive x value as $x_{k+1} = x_k + 1/m$ **……..(eq4)**

Note :

Equations 3 and 4 are based on the assumption that lines are to be **processed from the left endpoints to the right endpoint**. If this processing is reserved ie from the right endpoint to the left endpoint, then equations 3 and 4 become

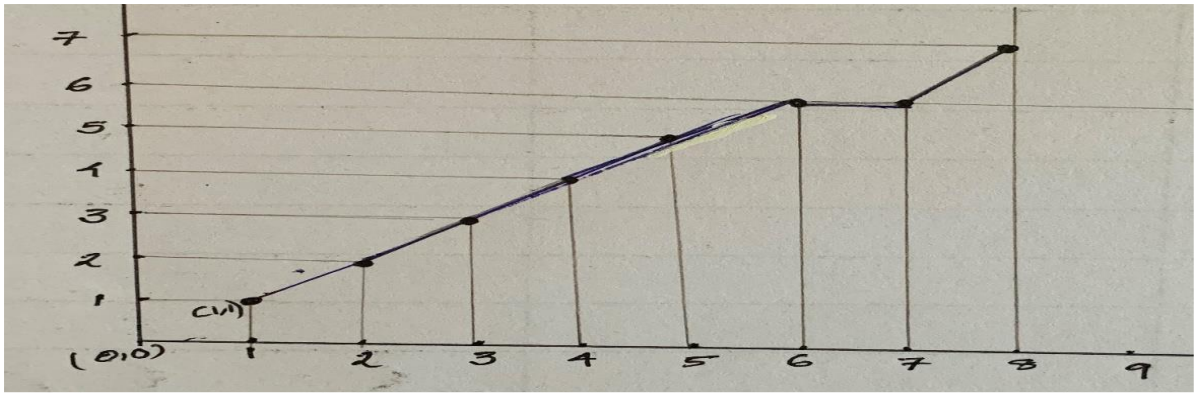
$y_{k+1} = y_k - m$ & $x_{k+1} = x_k - 1/m$

Draw line from (1,1) to(8,7) using the DDA algorithm

m=(7-1)/(8-1) = 6/7 =0.9 <1 then follow case 1

increment $x_{k+1}=x_k+1$ and $y_{k+1}=y_k+m$ initially plot (1,1)

No	Xk+1	Yk+1	Pixel Plotted Round(Xk+1 , Yk+1)
1	1+1=2	1+6/7=1.9	Round(2,1.9)=(2,2)
2	2+1=3	1.9 +6/7=2.8	Round(3,2.8)=(3,3)
3	3+1=4	2.8 + 6/7=3.7	Round(4, 3.7)=(4,4)
4	4+1=5	3.7+6/7 =4.6	Round(5, 4.6)=(5,5)
5	5+1=6	4.6 +6/7=5.5	Round(6, 5.5)=(6,6)
6	6+1=7	5.5 +6/7=6.4	Round(7, 6.4)=(7,6)
7	7+1=8	6.4 + 6/7=7.3	Round(8, 7.3)=(8,7)



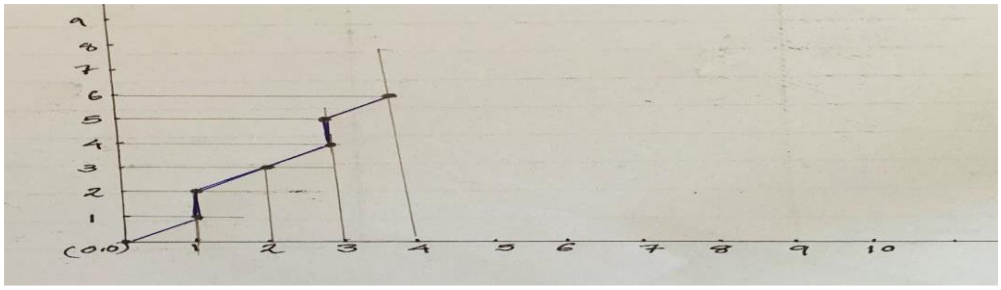
Draw line from (0,0) to(4,6) using the DDA algorithm

m=(6-0)/(4-0) = 6/4 =1.5>1 then follow case 2

increment $y_{k+1}=y_k+1$ and $x_{k+1}=x_k+1/m$

initially plot (0,0)

No	Xk+1	Yk+1	Pixel Plotted Round(Xk+1 , Yk+1)
1	0+4/6 =0.67	0+1=1	Round(0.67,1)=(1 ,1)
2	0.67+0.67=1.34	1+1=2	Round(1.34,2)=(1,2)
3	1.34+0.67=2.01	2+1=3	Round(2.01,3)=(2,3)
4	2.01+0.67=2.68	3+1=4	Round(2.68,4)=(3,4)
5	2.68+0.67=3.35	4+1=5	Round(3.35,5)=(3,5)
6	3.35+0.67=4.02	5+1=6	Round(4.02,6)=(4,6)



If this processing is reserved ie from the right endpoint to the left endpoint, then equations 3 and 4 become

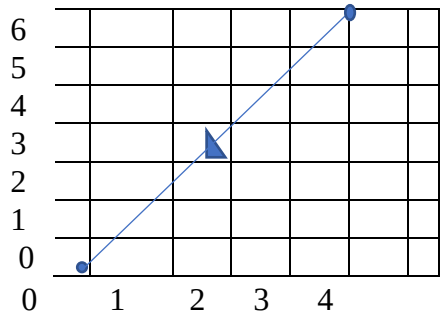
$$y_{k+1} = y_k - m \quad \& \quad x_{k+1} = x_k - 1/m$$

Draw line from (4,6) to(0,0) using the DDA algorithm

m=(0-6)/(0-4) = -6/-4 =1.5>1 then follow case 2

increment $y_{k+1}=y_k-1$ and $x_{k+1}=x_k-1/m$

initially plot (4,6)



o	Xk+1	Yk+1	Pixel Plotted Round(Xk+1 , Yk+1)
1	3.334	5	Round(3.334,5)=(3 ,5)
2	2.668	4	Round(2.668,4)=(3,4)
3	2.002	3	Round(2.002,3)=(2,3)
4	1.336	2	Round(1.336,2)=(1,2)
5	0.67	1	Round(0.67,1)=(1,1)
6	0.004	0	Round(0.004,0)=(0,0)

```
#include "device.h"
#define Round(a) ((int)(a+0.5))

void lineDDA(int xa, int ya, int xb, int yb)
  { int dx=xb-xa, dy=yb-ya , steps, k;
    float xIncrement , yIncrement , x=xa , y=ya;

   if(abs(dx)>abs(dy))
     steps=abs(dx);
  else
    steps=abs(dy);

  xIncrement=dx/(float)steps;
  yIncrement=dy/(float)steps;

 setPixel(Round(x) , Round(y));
for(k=0;k<steps;k++)

   {  x+=xIncrement;
      y+=yIncrement;
    setPixel(Round(x),Round(y));
  }
}
```

The advantages of DDA Algorithm are-

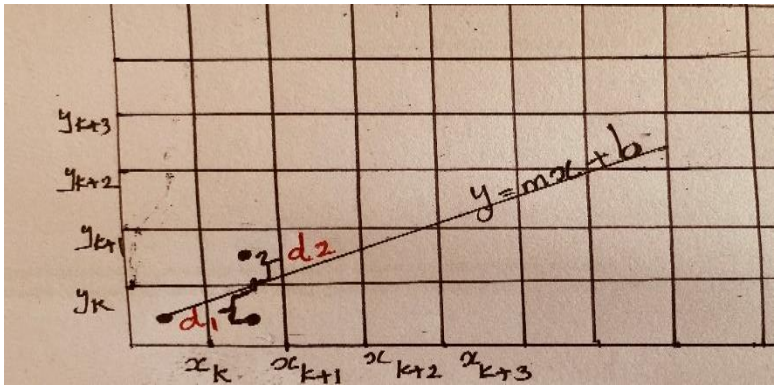
- It is a simple algorithm.
- It is easy to implement.
- It avoids using the multiplication operation which is costly in terms of time complexity.

The disadvantages of the DDA Algorithm are-

- There is an extra overhead of using the round-off () function.
- Using the round-off () function increases the time complexity of the algorithm.
- Resulted lines are not smooth because of the round-off () function.
- The points generated by this algorithm are not accurate.

Bresenham's Line Drawing Algorithm

An accurate and efficient raster line generating algorithm.



Line equation $y=m.x+b$ …….(1)

The y coordinate on the mathematical line at pixel column position x_k+1 is calculated as

$y=m(x_k+1)+b$ ……(2)

then $d1=y-y_k$

$= m(x_k+1)+b-y_k$ ……(3)

$d2=(y_k+1)-y$

$= y_k+1 – m(x_k+1)-b$ ……(4)

The difference between d1 and d2 is

$d1-d2 = m(x_k+1)+b-y_k - (y_k+1 – m(x_k+1)-b)$

$= 2\,m(x_k+1) - 2y_k +2b -1$ ……(5)

Substitute $m=dy/dx$ in equ 5

$(d1-d2)dx = 2.dy(x_k+1) -2y_k.dx+2b.dx -dx$

$(d1-d2)dx = 2.dy\ x_k +2.dy-2y_k.dx+2b.dx -dx$

$p_k = \mathbf{2.dy\ x_k – 2dx.y_k} +2.dy+2b.dx -dx$

$p_k = \mathbf{2.dy\ x_k – 2dx.y_k + C}$ ……..**(6)** where P_k is decision parameter

if($pk <=0$ ie d1-d2 <0 ,meaning actual line lays closer to yk ,then next pixel coordinate is (x_k+1,y_k) .

if($pk >0$ ie d1-d2 >0 ,meaning actual line lays closer to yk+1 ,then next pixel coordinate is (x_k+1,y_k+1) .

Calculate successive decision parameters

At step k+1 the decision parameter is $p_{k+1}=2.dy.x_{k+1}-2dxy_{k+1}+C$ ……(8)

equ(8)-equ(6)

$$p_{k+1} - p_k = 2.dy.x_{k+1} - 2dxy_{k+1} + C - (2.dy\ x_k - 2dx.y_k + C)$$
$$= 2.dy(x_{k+1} - x_k) - 2.dx(y_{k+1} - y_k)$$
$$\mathbf{p_{k+1} = p_k + 2.dy(x_{k+1} - x_k) - 2.dx(y_{k+1} - y_k) \ldots\ldots\ldots\ldots..(9)}$$

if pk <=0 then next pixel coordinate is (x_k+1, y_k) and $\mathbf{p_{k+1} = p_k + 2.dy(x_k+1-x_k)}$

$$\mathbf{p_{k+1} = p_k + 2.dy \ \ldots\ldots\ldots(10)}$$

If pk >0 , then next pixel coordinate is (x_k+1, y_k+1) and

$$\mathbf{p_{k+1} = p_k + 2.dy(x_k+1-x_k) - 2.dx(y_k+1-y_k)}$$
$$\mathbf{p_{k+1} = p_k + 2.dy - 2.dx \ \ldots\ldots\ldots..(11)}$$

The first decision parameter P0 is evaluated at starting pixel position (x0,y0) and with m evaluated as dy/dx ,using equation (6)

$$p_k = \mathbf{2.dy\ x_k - 2dx.y_k} + 2.dy + 2b.dx - dx$$

y=mx+b , b=y-mx →b=y-(dy/dx).x substitute in above equation

$$p_0 = \mathbf{2.dy\ x_0 - 2dx.y_0} + 2.dy + 2(y_0 - (dy/dx)x_0).dx - dx$$

$$p_0 = \mathbf{2.dy\ x_0 - 2dx.y_0} + 2.dy + 2y_0.dx - 2dy.x_0 - dx$$

$$\mathbf{p_0 = 2.dy - dx \ \ldots\ldots\ldots\ldots\ldots..(12)}$$

Bresenham's Line-Drawing Algorithm for |m|<1

1. Input the two endpoints and store the left endpoint in (x0,y0).

2. Load (x0,y0) into the frame buffer; that is plot the first point.

3. Calculate the constants dx, dy, 2dy, and 2dy–2dx and get the first value for the decision parameter as

$$\mathbf{p0 = 2dy - dx}$$

4. At each X_k along the line, starting at k = 0, perform the following test –

If **pk < 0,** the next point to plot is **(x_k+1, y_k)** and **$P_{k+1} = P_k + 2dy$**

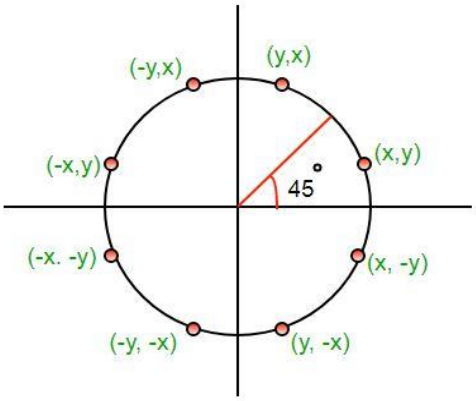
Otherwise, the next point to plot is **(x_k+1, y_k+1)** and **$P_{k+1} = P_k + 2dy - 2dx$**

5. Repeat step 4 dx times.

Mid-Point Circle Drawing Algorithm

The **mid-point** circle drawing algorithm is an algorithm used to determine the points needed for rasterizing a circle.

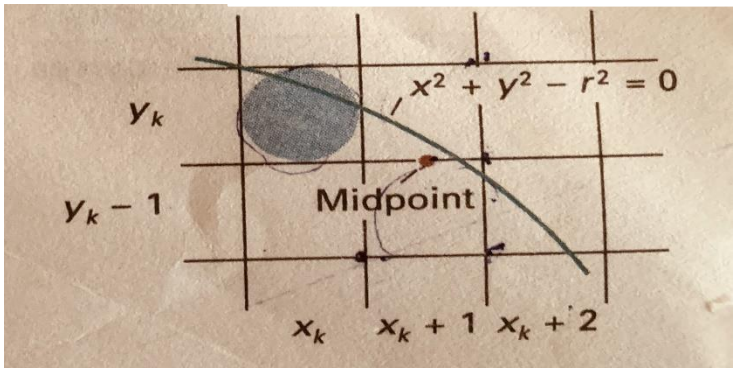
We use the **mid-point** algorithm to calculate all the perimeter points of the circle in the **first octant** and then print them along with their mirror points in the other octants. This will work because a circle is symmetric about its center.



Consider 8-way circle symmetry and find all points in a single octant and corresponding to each point find out all other seven points of other octants, so after covering a single octant we will get all circle points.

For a given radious "r" and circle centre (xc,yc) ,we can first set up our algorithm to calculate pixel positions around a circular path centered at the coordinate origin(0,0). Then each calculated position (x,y) is moved to its proper screen position by adding xc to x and yc to y.
We process one octant which is present in the first quadrant and varies from x=0 to x=y.
Circle function $f_{circle}(x,y)=x^2+y^2-r^2$ is taken as the decision parameter .

$\mathbf{f_{circle}(x,y)=}$
- <0 ,if(x,y) is inside the circle boundary
- =0 , if(x,y) is on the circle boundary
- >0 , if(x,y) is outside the circle boundary



Decision making

Assuming we have plotted the pixel at (x_k,y_k),we need to determine the next pixel at (x_{k+1},y_{k+1}). Choose one pixel from (x_k+1,y_k) ,(x_k+1,y_k-1) which is closest to the circle boundary . use circle function $f_{circle}(x,y)=x^2+y^2-r^2$ as the decision parameter and take midpoint of $(x_k+1,\ y_k)$ and (x_k+1,y_k-1) as its x and y values.

$P_k = \mathbf{f_{circle}(x_{k+1},y_{k-1/2})}$
$= (x_k+1)^2 +(y_k-1/2)^2-r^2$….(1)

Midpoint of x_k+1 , x_k+1

$(x_k+1+ x_k+1)/2 = (2 .x_k+1)/2$

$= x_k+1$

Midpoint of y_k,y_k-1

$(y_k +y_k-1)/2 = (2y_k-1)/2$

$=y_k -1/2$

If $p_k<0$, this midpoint is inside the boundary, and we select y_k..
Otherwise, the midpoint is outside or on the circle boundary and we select the pixel on scan line y_k-1.
Successive decision parameters are obtained using incremental calculations.
We obtain a recursive expression for the next decision parameter by evaluating the circle function at the sampling position $(x_{k+1}+1,y_{k+1}-1/2)$

$P_{k+1} = (x_{k+1}+1)^2 + (y_{k+1}-1/2)^2-r^2$

$= (x_k+1+1)^2 + (y_{k+1}-1/2)^2-r^2$

$= (x_k+2)^2 + (y_{k+1}-1/2)^2-r^2$

$= x_k^2 +4\,x_k +4 + (y_{k+1})^2 -(y_{k+1}) +1/4- r^2$

$P_k = (x_k+1)^2 +(y_k-1/2)^2-r^2$

$= x_k^2 +2\,x_k +1+ (y_k)^2 -(y_k) +1/4- r^2$

$P_{k+1}- P_k = x_k^2 +4\,x_k +4 + (y_{k+1})^2 -(y_{k+1}) +1/4- r^2 -(x_k^2 +2\,x_k +1+ (y_k)^2 -2(y_k) +1/4- r^2)$

$=2\,x_k+3+ (y_{k+1})^2-(y_{k+1})- (y_k)^2+(y_k)$

$\mathbf{P_{k+1} =P_k+2x_k+3+ (y_{k+1})^2-(y_{k+1})- (y_k)^2+(y_k)}$

If $p_k<0$,then point (x_k+1, y_k)

$\mathbf{P_{k+1}} = P_k+2\,x_k+3+y_k^2-y_k- y_k^2+y_k$

$\mathbf{= P_k+2\,x_k+3}$

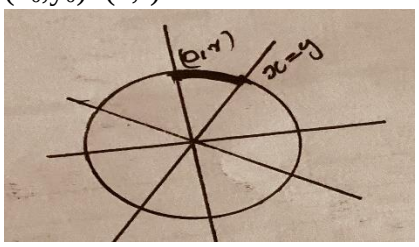
If $p_k>0$,then point (x_k+1, y_k-1)

$\mathbf{P_{k+1}} =P_k+2\,x_k+3+ (y_{k-1})^2-(y_{k-1})- (y_k)^2+(y_k)$

$=P_k+2\,x_k+3+ y_k^2-2\,y_k +1-y_k+1- y_k^2+y_k$

$\mathbf{=P_k+2\,x_k-2\,y_k+5}$

Initial Decision parameter is obtained by evaluating the circle function at the position $(x_0,y_0)=(0,r)$



$P_0 = f_{circle}(1,r-1/2)$

$=1+(r-1/2)^2-r^2$

P_0=5/4 -r,if the radius r is specified as an integer, we can simply round p_0 to **p_0=1-r .**

Midpoint Circle Algorithm

1. Input radius r and circle center (x_0,y_0) ,and obtain the first point on the circumference of a circle centered 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**
3. At each x_k position , strting at k=0 , perform the following test : if $p_k<0$, the next point along the circle centered on (0,0) is(x_{k+1},y_k) and
   $P_{k+1} = P_k+2x_k+3$
   Otherwise next point along the circle is (x_{k+1},y_k-1) and
   $P_{k+1} = P_k+2x_k-2y_{k+} 5$
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 coordinate values :
   $x=x+x_c$ $y=y+y_c$

Example

Given a circle radius r=10 , determine positions along the circle octant in the first quadrant from x=0 to x=y. The initial value of the decision parameter is

P0=1-r=1-10=-9

Initial point is $(x_0,y_0)=(0,10)$

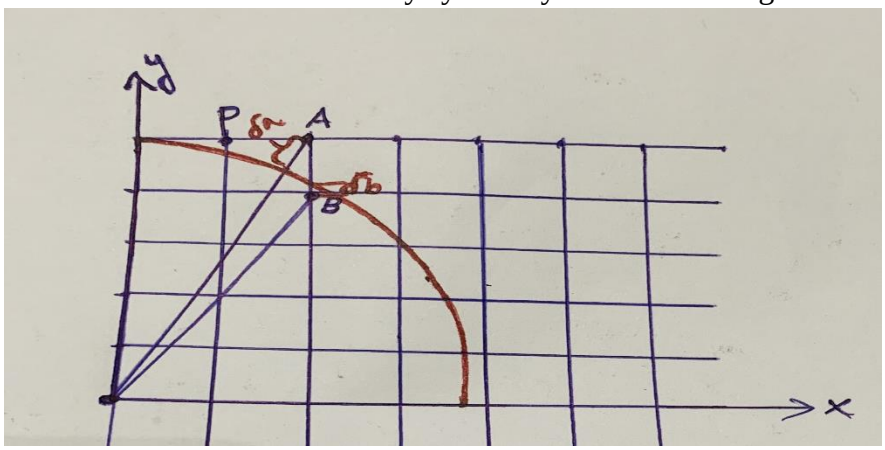
$2x_0=0$ $2y_0=20$

k	pk	(x_{k+1},y_{k+1})	$2x_k+3$	$2x_k-2y_k+5$
0	-9	(1,10)	3	
1	-9+3=-6	(2,10)	5	
2	-6+5=-1	(3,10)	7	
3	-1+7=6	(4 , 9)		6-20+5=-9
4	6-9= -3	(5,9)	11	
5	-3+11 =8	(6 ,8)		10-18+5 =-3
6	8 -3 =5	(7,7) →x=y ,then stop		

Corresponding to each obtained point, find out 7 other symmetric points also in other 7 octants and draw the complete circle.

Bresenham's Circle Drawing Algorithm.

It is an efficient method for circle drawing. It avoids the trigonometric and square root calculations by adopting only integer operations involving the square of the pixel separation distances. It considers the 8-way symmetry of the circle to generate a complete circle.



Distance of pixel A from the origin (0,0)

$D_A=\sqrt{(xi+1)^2+yi^2}$

Distance of pixel B from the origin (0,0)

$D_B=\sqrt{(xi+1)^2+(yi-1)^2}$

The distance of pixel A and B from the true circle are

da=DA-r db=DB-r

To avoid square root take a square

$da=DA^2-r^2$ $db=DB^2-r^2$

$da=(xi+1)^2+yi^2-r^2$ $db=(xi+1)^2+(yi-1)^2-r^2$

Decision Variable **di=da+db**

$di = 2(x_i+1)^2+2y_i^2-2y_i-2r^2+1$

Successive Decision parameter

$d_{i+1}= 2(x_{i+1}+1)^2+2y_{i+1}^2-2y_{i+1}-2r^2+1$

$=2(x_i+2)^2 +2y_{i+1}{}^2-2y_{i+1}-2r^2+1$

$d_{i+1}- di =2(x_i+2)^2 +2y_{i+1}{}^2-2y_{i+1}-2r^2+1-(2(x_i+1)^2 +2y_i{}^2-2y_i-2r^2+1)$

$d_{i+1}=d_i +4x_i+2y_{i+1}{}^2-2y_{i+1}-2y_i^2+2y_i+6$

Case1: di<0

$x_{i+1}=x_i+1$

$y_{i+1}=y_i$

$d_{i+1}=d_i +4x_i+2y_i{}^2-2y_i-2y_i^2+2y_i+6$;

$d_{i+1}=d_i +4x_i+6$

Case2: di>=0

$x_{i+1}=x_i+1$

$y_{i+1}=y_i -1$

$d_{i+1}= d_i +4x_i+2(y_i-1)^2-2(y_i-1)-2y_i^2+2y_i+6$

$d_{i+1}=d_i+4x_i+2y_i^2-2y_i+2-2yi+2-2y_i^2+2y_i+6$

$d_{i+1}=d_i +4x_i -4y_i+10$

Initial Decision Parameter

Initial Pixel at (0,r)

$di = 2(x_i+1)^2 +2y_i{}^2-2y_i-2r^2+1$

$di = 2(0+1)^2 +2r^2-2r-2r^2+1$

$=2+2r^2-2r-2r^2+1$

d0 =3-2r

Bresenham's Circle Drawing Algorithm

1. Input radius r and circle center(x_c,y_c) , then set the coordinates for the first point on the circumference of a circle centered on the origin as $(x_0,y_0)=(0,r)$.
2. Calculate the initial decision parameter as d0=3-2r
3. At each x_i , from i=0 perform the following :
   If $d_i<0$,next point to plot along the circle centered on (0,0) is (x_{i+1},y_i) and
   $d_{i+1}=d_i+4x_i+6$.
   Otherwise ,
   Next point to plot is (x_{i+1},y_{i-1}) and
   $d_{i+1}=d_i +4(xi-yi)+10$.
4. Determine the symmetry points in the other seven octants.
5. Move each calculated pixel position (x,y) onto the circular path centered at (xc,yc) and plot the coordinate values: $x=x+x_c$; $y=y+y_c$;
6. Repeat steps 3 through 5 until x>=y.

Example

Given a circle radius r=10 , determine positions along the circle octant in the first quadrant from x=0 to x=y. The initial value of the decision parameter is

d0=3-2x10=-17

Initial point is $(x_0,y_0)=(0,10)$

k	dk	(x_{k+1},y_{k+1})	$4x_i+6$	4(xi-yi)+10
0	-17	(1,10)	4x0+6=6	
1	-17+6=-11	(2,10)	4x1+6=10	
2	-11+10=-1	(3,10)	4x2+6=14	
3	-1+14=13	(4,9)		4(3-10)+10=-18
4	13-18=-5	(5,9)	4x4+6=22	
5	-5+22=17	(6,8)		4(5-9)+10=-6
6	117-6=11	(7,7) x=y then stop		

University questions…

- Distinguish between raster scan display and random scan display.
- What do you understand about the aspect ratio and resolution of a display screen in a raster scan display?
- Explain the working of a random scan display system with a suitable diagram.
- Explain the working of a beam penetration CRT.
- Explain the working of direct view storage tubes (DVST).
- Consider a raster system with a resolution of 2560*2048. Determine the frame buffer size (in bytes) needed for the system to store12-bits per pixel. How much storage is required if 24-bits per pixel are to be stored?
- Explain the working of a delta-delta shadow mask CRT.
- What is the use of computer graphics? List out and explain any 5 applications of computer graphics.
- Draw and explain the architecture of simple raster graphics systems.
- Define the following terms.
  Persistence.
  Resolution.
  Aspect ratio.
  Framebuffer.
- Describe in detail the basic video controller refresh operation used in interactive raster graphics systems.
- Explain the working of the cathode ray tube in detail.
- What is the role of a display controller in a raster scan display system? Explain.
- Explain the functioning of a random scan display system.
- With a neat diagram describe the working of a cathode ray display device.
- Write the methods used to plot a dashed line segment.
- Using the midpoint circle drawing algorithm find out the first quadrant point of a circle from x=0 to x=y where r=10. Also, draw the complete circle using the same.
- Using both DDA and Bresenhams's line drawing algorithm, plot the line with endpoints (20,15) and (34,20).
- How 8-way symmetric points if (x,y) is a point on the circle with centre at origin.
- Write the DDA line drawing algorithm.
- Rasterize the line segment from pixel coordinate(1,1) to (8,5) using Bresenham's line drawing algorithm.
- Write the midpoint circle drawing algorithm.
- Use midpoint circle drawing algorithm to plot a circle whose radius=20 units and center is (50,30).
- What are the advantages and disadvantages of the DDA algorithm?
- Describe the DDA line drawing algorithm. Use the algorithm to find the coordinate along the line joining the pixel positions (5,12) and (15,20).
- Explain DDA line drawing algorithm with examples.
- Explain the midpoint circle drawing algorithm. Find the pixel locations approximately the first octant of a circle having centre(10,13) and radius of 5units using this algorithm.
- Describe simple random scan display system and draw its architecture.
- With a suitable figure, describe the shadow masking techniques in CRT.
- Compare the working principle of raster scan systems and random scan systems.

- How much time is spent scanning across each row of pixels during screen refresh on a raster system with a resolution of 1280*1024 and are the fresh rate of 60 frames per second?
- Rasterize the line with endpoints (2,3) and (5,8) using Bresenham’s line drawing algorithm.