CG- MODULE 1

- The term **computer graphics** includes almost everything on computers that is not text or sound.
- Computer graphics is art of drawing pictures on computers, also called as rendering.
- Computer Graphics express data in pictorial form.
- It displays information in the form of graphics objects such as pictures, charts ,graphs etc instead of simply text.
- Computer graphics is an art of drawing pictures, lines, charts, etc using computers with the help of programming.

Applications of computer graphics

- Movie industry
- Games
- Medical Imaging and Scientific Visualization
- Computer Aided Design
- Education & training
- Simulators used for training ship captains,pilots
- Computer Art
- Presentation graphics
- Image processing
- GUI

- A display area is represented as a collection of discrete picture elements called pixels.
- Pixel is the smallest addressable unit on the screen. Each pixel has got some information like its intensity and color.
- Basically there are two types of computer graphics namely Interactive Computer Graphics and Passive computer graphics

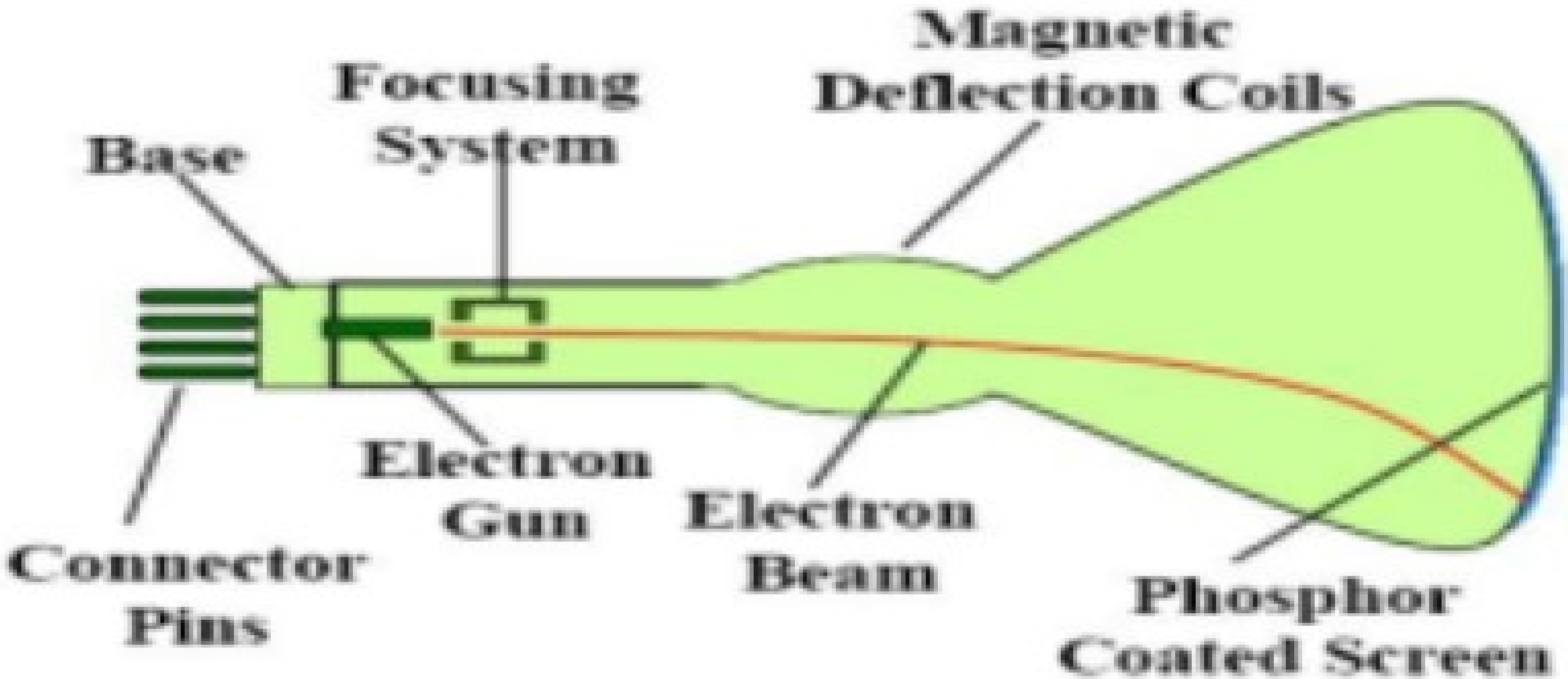
Interactive Computer Graphics

- It involves a two way communication between computer and user.
- The observer is given some control over the image by providing him with an input device .
- For example: Observer send his request to the computer through video game controller. Computer on receiving signals can modify the displayed picture appropriately.

Non Interactive Computer Graphics

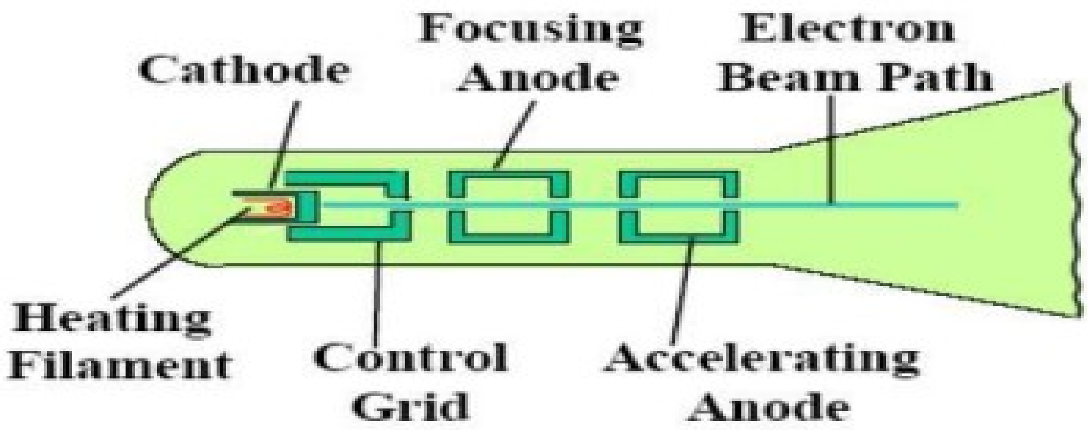
- It is also known as passive computer graphics.
- The user does not have any kind of control over the image.
- It involves only a one way communication.
- The image is merely the product of static stored program and will work according to the instructions given in the program linearly.
- The image is totally under the control of program instructions, not under the user. Example: images shown on television.

Video Display Devices: CRT



- The primary output device in a graphics system is a video monitor.
- The operation of most video monitors is based on the standard cathode ray tube (CRT).

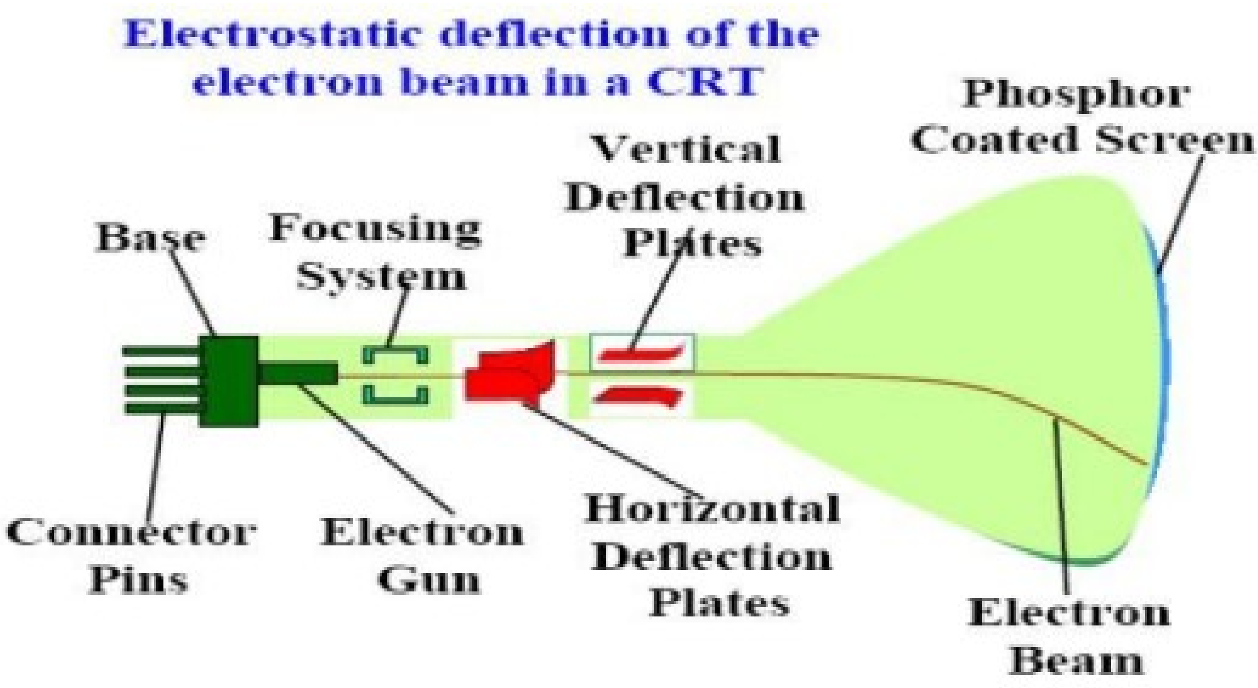
- The beam of electrons are emitted by an electron gun. It passes through focusing and deflection systems that direct the beam toward specified positions on the phosphor coated screen.
- Thus phosphor emits a small spot of light at each position contacted by the electron beam.
- Because the light emitted by the phosphor fades very rapidly, some method is needed for maintaining the screen picture.
- 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
- 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. This causes electrons to be ‘boiled off" the hot cathode surface.
- Intensity of the electron beam is controlled by setting voltage levels on the control grid, which is a metal cylinder that fits over the cathode.
- Amount of light emitted by the phosphor coating depends on the number of electrons striking the screen, thus we control the brightness of a display by varying the voltage on the control grid.

- A smaller negative voltage on the control grid simply decreases the number of electrons passing through the small hole at the end of the control grid structure.
- A high negative voltage applied will stop electrons from passing through it.
- The focusing system force the electron beam to converge into a small spot as it strikes the phosphor. Otherwise, the electrons would repel each other, and the beam would spread out as it approaches the screen.
- Focusing is accomplished with the help of either electric or magnetic fields.
- It is commonly constructed with a pair of magnetic deflection coils mounted on CRT envelope. The magnetic field produced by each pair of coils directs the electron beam properly. Horizontal deflection is accomplished with one pair of coils, and vertical deflection by the other pair.

- When electrostatic deflection is used, two pairs of parallel plates are mounted inside the CRT envelope. One pair of plates is mounted horizontally to control the vertical deflection, and the other pair is mounted vertically to control horizontal deflection.



Persistence

- Persistence is defined as the time CRT takes to decay the emitted light from the screen to one tenth of its original intensity. Lower persistence phosphor require high refresh rate. Graphics monitors are usually constructed with a persistence in the range from 10 to 60 microseconds.

Resolution

- The maximum number of points that can be displayed without overlap on a CRT. Higher resolution systems are

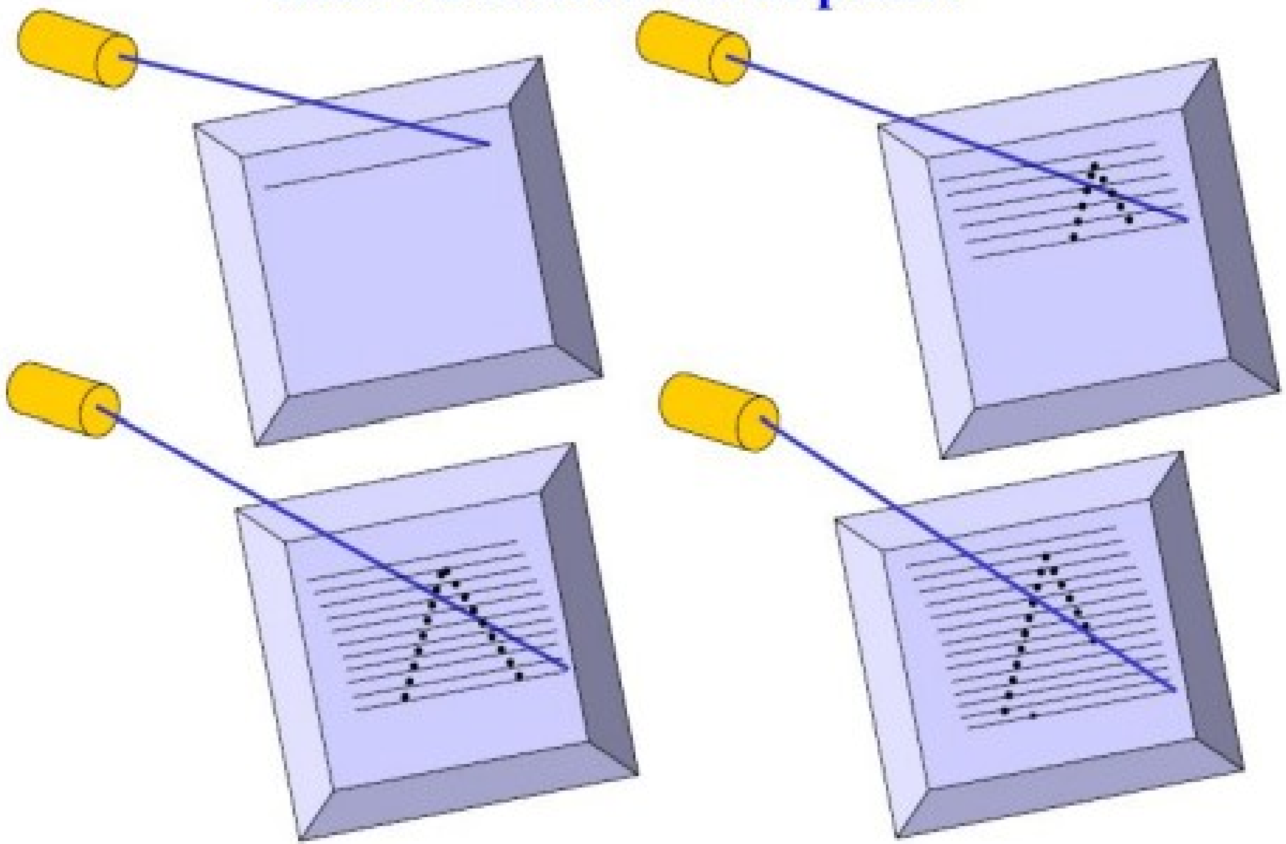
Aspect Ratio :

- The ratio of vertical points to horizontal points necessary to produce equal length lines in both direction on the screen. It is the ratio of horizontal to vertical points.

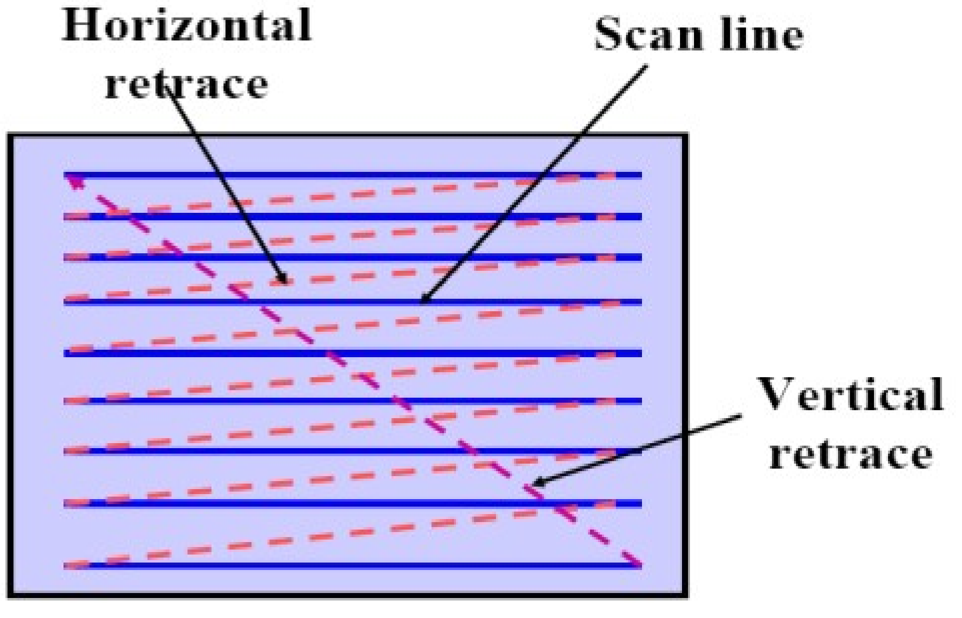
Raster scan display

- It is the most common type of CRT monitor.
- In this system, the electron beam is swept across the screen one row at a time from top to bottom.
- Beam intensity is turned on and off to create a pattern of illuminated spots as the electron moves across each row.
- In a raster scan system entire screen is considered as a matrix of pixels.
- Each screen point is referred to as a pixel or pel (shortened form of picture element).
- Frame buffer or refresh buffer is a memory where picture definition is stored .i.e. set of all intensity values for all pixels is stored in refresh buffer.

Raster-scan display system draws a discrete set of points



- **Horizontal Retrace**: At the end of each scan line , the electron beam returns to the left side of the screen to begin displaying the next scan line.
- **Vertical retrace**: At the end of each frame the electron beam returns to the top left corner of the screen to begin the next frame.

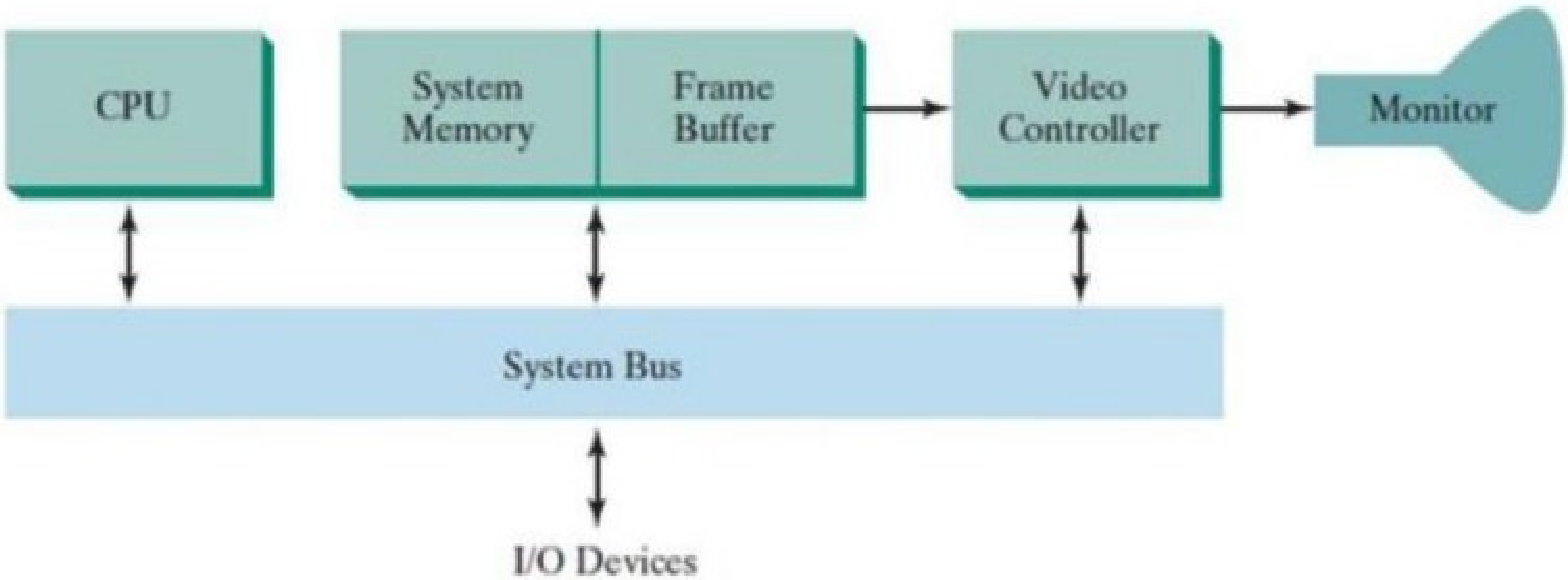


- **Interlacing**: In this method, each frame is displayed in two passes using an interlaced refresh procedure. First all the points on the even numbered scanlines are displayed ,and then all the odd numbered lines are displayed. Entire screen is displayed in half time , providing that adjacent scanlines contain similar information.

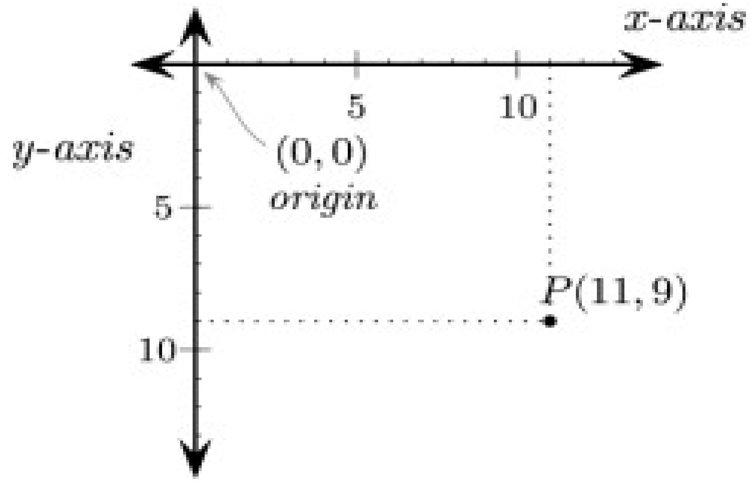
- In a simple black and white system, each screen point is either on or off, so only one bit per pixel is needed to control the intensity of screen positions.
- Bit value of 1 indicates that the electron beam is to be turned on that position, and a value of 0 indicates that the beam intensity is to be turned off.
- Additional bits are needed when color and intensity variations are to be displayed.
- On a black-and-white system with one bit per pixeI, the frame buffer is commonly called a bitmap.
- For systems with multiple bits per pixel, the frame buffer is referred to as a pixmap.
- Refreshing on raster scan displays is carried out at the rate of 60 to 80 frames per second.
- Home television sets and printers are examples of other systems using raster-scan methods.

RASTER-SCAN SYSTEMS

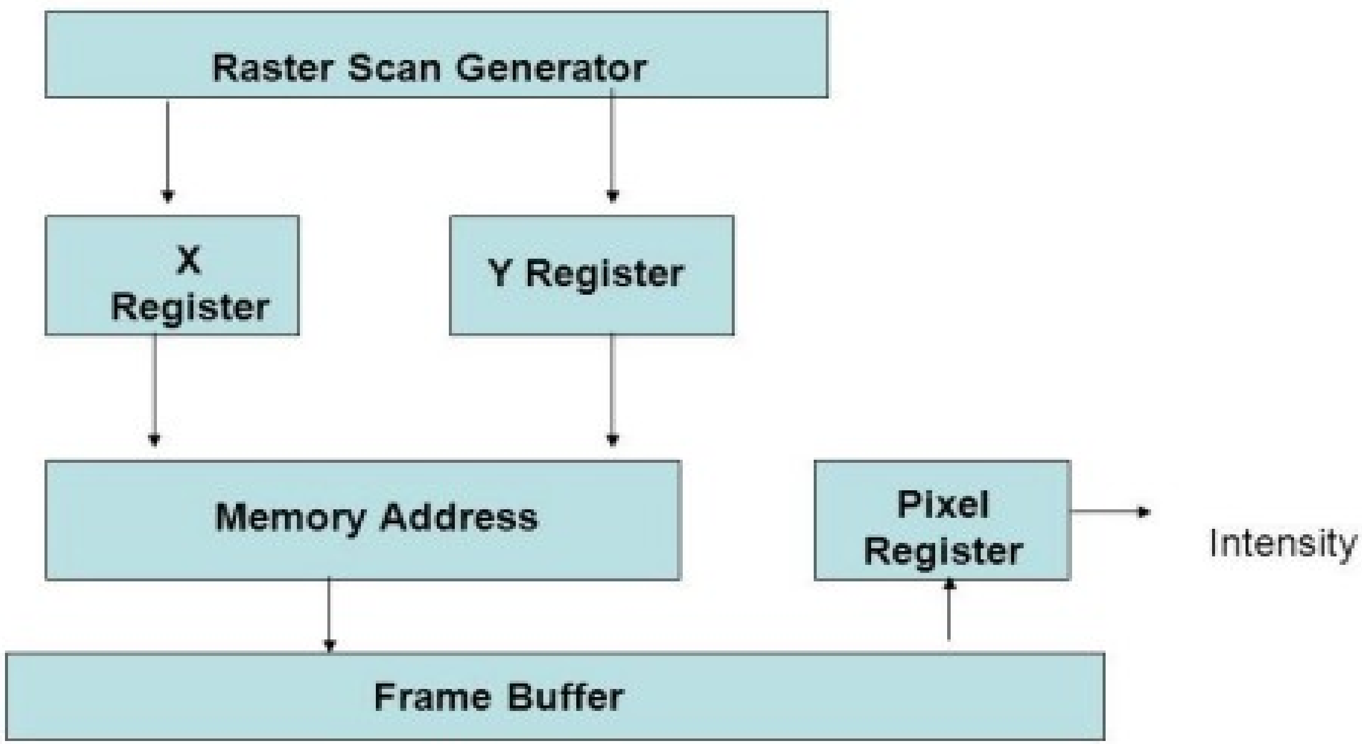
- In Interactive raster graphics systems, in addition to the CPU ,a special-purpose processor, called the video controller or display controller, is used to control the operation of the display device.



- 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 the corresponding screen positions, are referenced in Cartesian coordinates.
- For many graphics monitors, the coordinate origin is defined at the lower left screen corner.
- The screen surface is represented as the first quadrant of a two-dimensional system, with positive x values increasing to the right and positive y values increasing from bottom to top.
- Scan lines are then labeled from y_{max} at the top of the screen to 0 at the bottom.
- Along each scan line, screen pixel positions are labeled from 0 to x_{max}.
- On some personal computers, the coordinate origin is referenced at the upper left comer of the screen, so the y values are inverted.



Basic refresh operations of the video controller



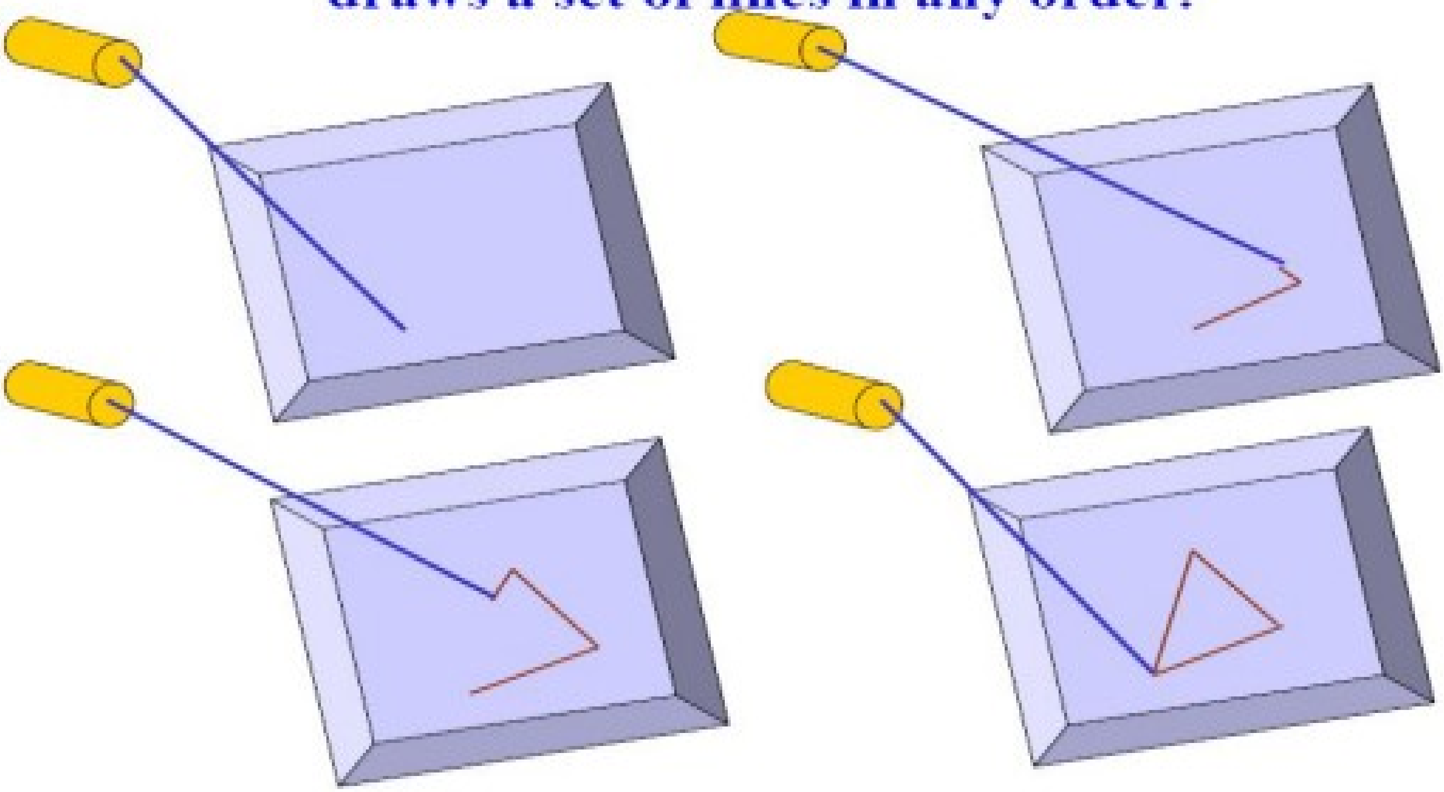
- Two registers are used to store the coordinates of the screen pixels: x register and y register.
- Initially, x register is set to 0 and the y register is set to y_{max}.
- The value stored in the frame buffer for this pixel position is then retrieved and used to set the intensity of the CRT beam.
- Then the x register is incremented by 1, and the process 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 scan 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.

- 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 called pixel register and used to control the CRT beam intensity for a group of adjacent pixels.
- In addition to the system memory, a separate display processor memory area can also be provided. Main purpose 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.

Random scan display

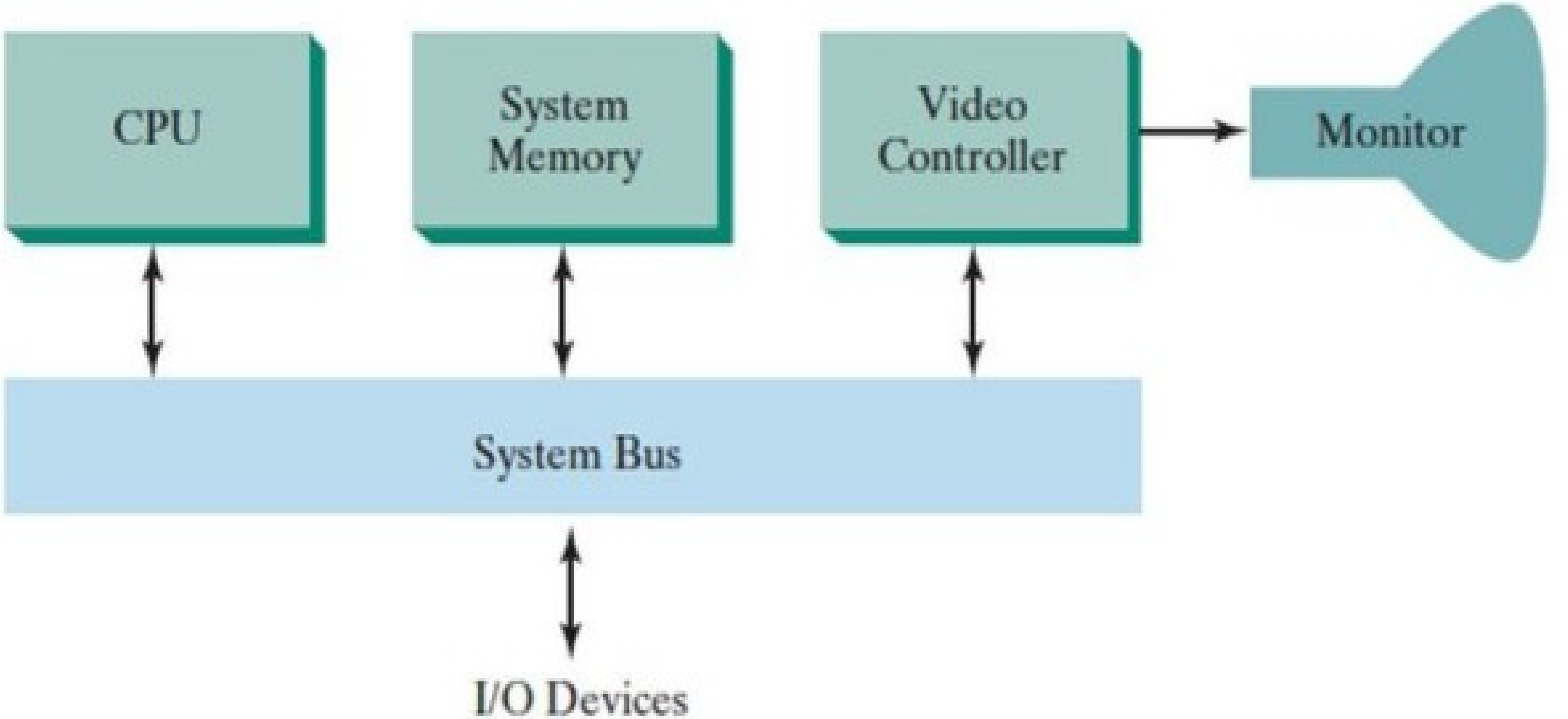
- It is also called as calligraphic displays, vector displays, stroke displays.
- The electron beam is directed only to the parts of the screen where a picture is to be drawn.
- Random scan monitors draw a picture one line at a time.
- A pen plotter operates in a similar way and is an example of a random-scan, hard-copy device.
- Picture definition is stored as a set of line drawing commands in an area of memory referred to as refresh display file/refresh buffer.
- To display a specified picture ,the system cycles through a set of commands in the display file , drawing each component line.
- After all line drawing commands have been processed, the system cycles back to the first line command in the list.
- Random-scan displays are scan displays are designed to draw all the component lines of a picture 30 to 60 times

Random-scan display system draws a set of lines in any order.



Raster displays	Random displays
Picture definition stored as set of intensity values	Picture definition stored as Line commands
Refreshing rate: 60 to 80 frames/sec	Refreshing rate: 30 to 60 frames/sec
Beam directed as continous scan line	Beam directed only to parts of image to be drawn
lower resolution	higher resolution
Draw Jagged lines	Draw Smooth lines
Example: printers	Example:plotters

RANDOM-SCAN SYSTEMS



- An application program is input and stored in the system memory along with a graphics package.
- Graphics commands in the application program are translated by the graphics package into a display file stored in the 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 once during every refresh cycle.
- Sometimes the display processor in a random-scan system is referred to as a display processing unit or a graphics controller.
- Graphics patterns are drawn on a random-scan system by directing the electron beam along the component lines of the picture.
- Lines are defined by the values for their coordinate endpoints, and these input coordinate values are converted to x and y deflection voltages.
- A scene is then drawn one line at a time by positioning the beam to fill in the line between specified endpoints.

LINE DRAWING ALGORITHMS

- There are two algorithms for drawing a line

✓ DDA line drawing algorithm

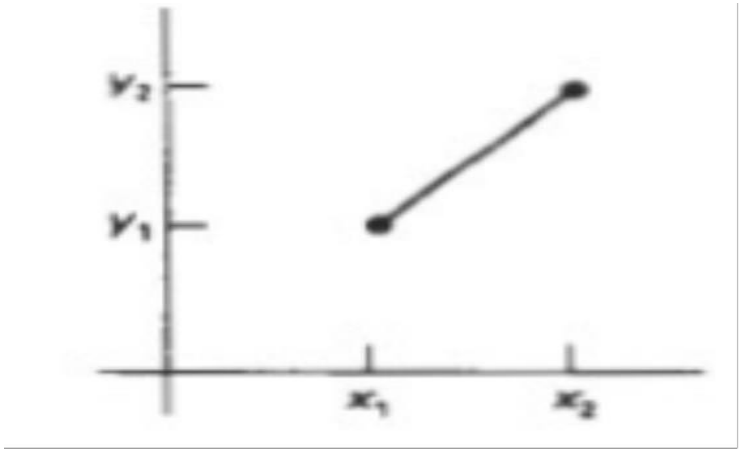
✓ Bresenham’s line drawing algorithm

Principle used for line drawing algorithms

- The Cartesian slope intercept eqn for straight line is $y = m x + b \longrightarrow$ (1)

with **m** representing the slope of the line and b as the **y** intercept.

- Given that the two endpoints of a line segment are specified at positions (x 1,y1) and (x2, y2), as shown in Figure.



- We can determine the slope m and y intercept b

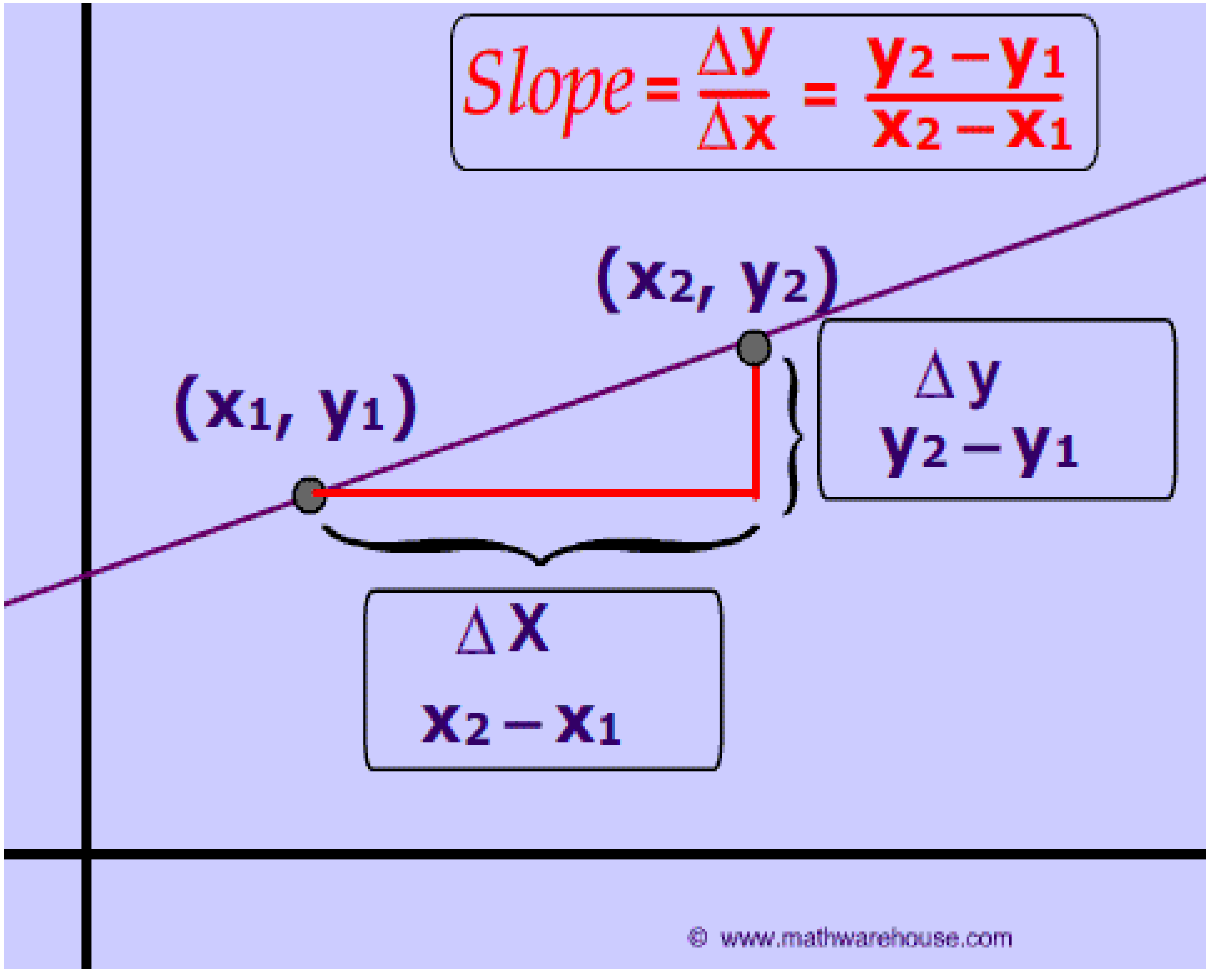
$$m=(y2-y1)/(x2-x1) \text{ ie } \Delta y/\Delta x \longrightarrow 2$$

$$b= y1- mx1 \longrightarrow 3$$

i.e

$$y1 = mx1 + b$$
$$y2 = mx2 + b$$

Algorithms for displaying straight lines are based on the line equation (1) and the calculations given in Eqs. (2)


$Slope = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$
(x_2, y_2)
(x_1, y_1)
Δy
$y_2 - y_1$
ΔX
$x_2 - x_1$
© www.mathwarehouse.com

For any given x interval Δx along a line, we can compute the corresponding y interval from (2) as

$$\Delta y = m\Delta x \qquad \text{-(4)}$$

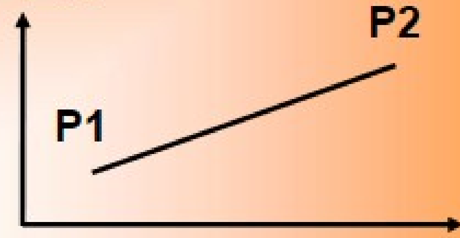
Similarly, we can obtain the x interval Δx corresponding to a specified Δy as

$$\Delta x = \frac{\Delta y}{m} \qquad \text{-(5)}$$

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

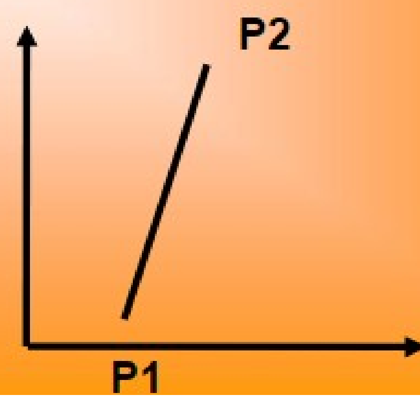
Lines with slope magnitude $|m| < 1$, Δx is set proportional to a small horizontal deflection voltage and the corresponding vertical deflection voltage Δy is $\Delta y = m\,\Delta x$

$\Delta x > \Delta y$



Lines with slope magnitude $|m| > 1$, Δy is set proportional to a small vertical deflection voltage and the corresponding horizontal deflection voltage is $\Delta x = \Delta y / m$

$\Delta y > \Delta x$



Digital Differential Algorithm(DDA)

- **Scan conversion algorithm based on calculating either Δx or Δy.**
- **Sample the line at unit intervals in one coordinate and determine the corresponding integer values nearest the line path for the other coordinate.**

Case 1

Line with positive slope less than or equal to 1, we sample at unit x intervals (Δx =1) and determine the successive y value as

$$Y_{k+1} = Y_k + m$$

$\Delta y = m \, \Delta x$

$Y_{k+1} - Y_k = m \, \Delta x$

$Y_{k+1} - Y_k = m \quad (\Delta x = 1)$

- k takes integer values starting from 1 and increases by 1 until the final point is reached.
- m can be any number between 0 and 1.
- Calculated y values must be rounded off to the nearest integer.
- For lines with positive slope greater than 1, sample at unit y intervals ($\sim y = 1$) and calculate each successive x value as

$$x_{k+1} = x_k + 1/m$$

The above eqns are based on the assumption that lines are processed from left to right. If the processing is reversed then

$\sim x = -1$ & $y_{k+1} = y_k - m$ Slope >1

$\sim Y = -1$ & $x_{k+1} = x_k - 1/m$ Slope <1

ALGORITHM

Step1: Input two end point pixel positions (x1, y1) and (x2, y2)
Step2: Find horizontal and vertical difference between the end points

dx = x2-x1

dy = y2-y1

Step3: The difference with the greater magnitude determines the value of parameter steps

If abs (dx)>abs (dy) then steps = abs (dx)

else

steps = abs (dy)

Step4: Starting with pixel position (x1, y1) be determined offset needed at each step to generate next pixel along the line path.

xincrment = dx/steps yincrement =dy/steps

step5: Loop through this process steps times.

x = x + xincrement

y = y + yincrement

Step6:Plot the pixel at (x, y) position, set pixel (round(x), round(y).

Advantages

- It is the simplest algorithm and it does not require special skills for implementation.
- It is a faster method for calculating pixel positions than the direct use of equation y=mx + b.

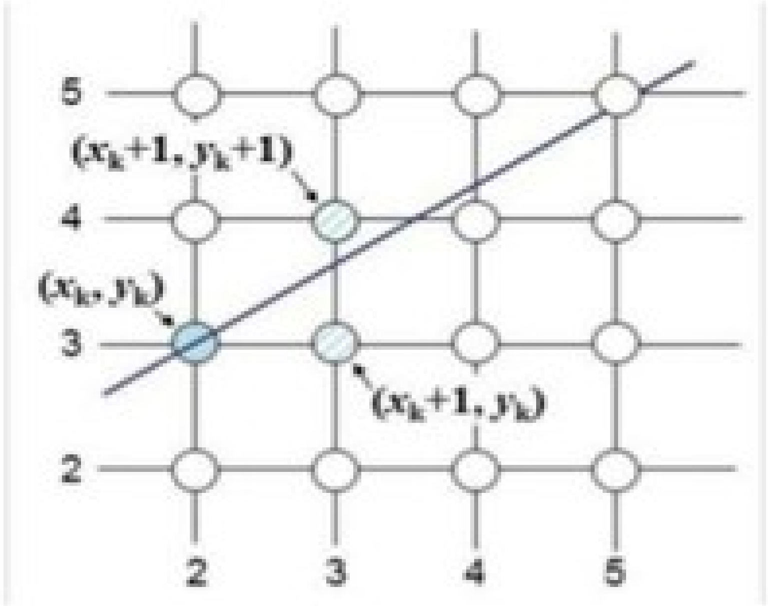
Disadvantages:

(1) The rounding operation & floating point arithmetic are time consuming procedures.

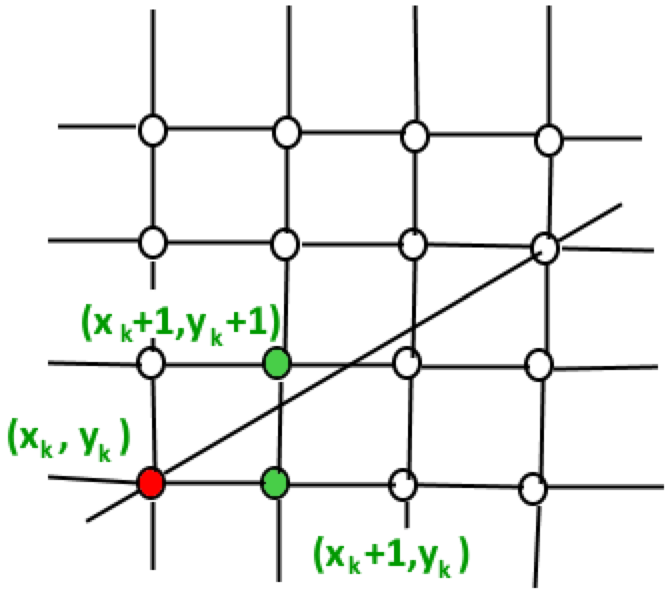
Perform integer arithmetic by storing float as integers in numerator and denominator and performing integer arithmetic

(2) Round-off error can cause the calculated pixel position to drift away from the true line path for long line segment. Hence end point accuracy is poor.

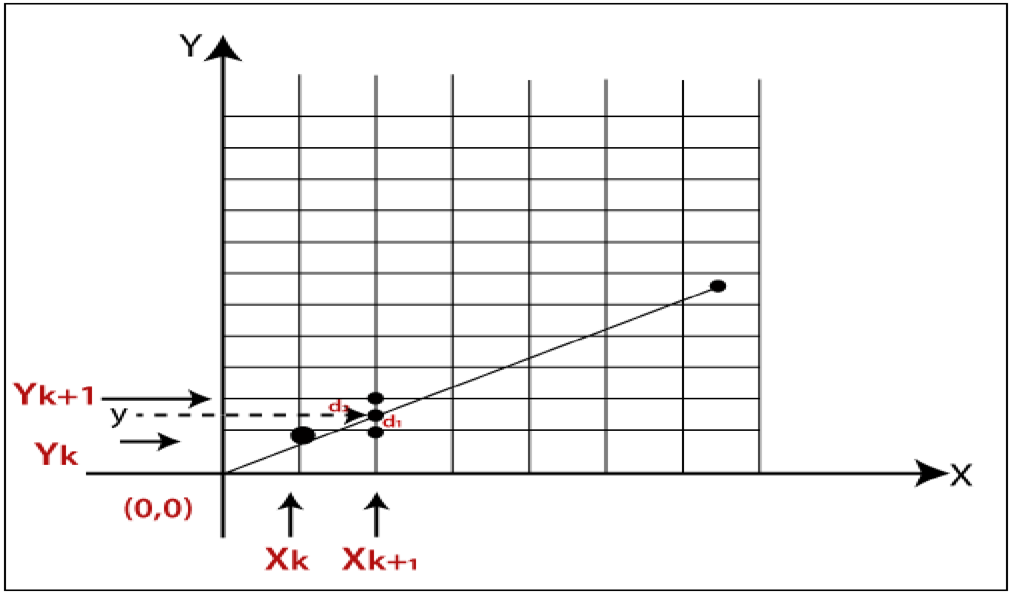
Bresenham's line drawing algorithm



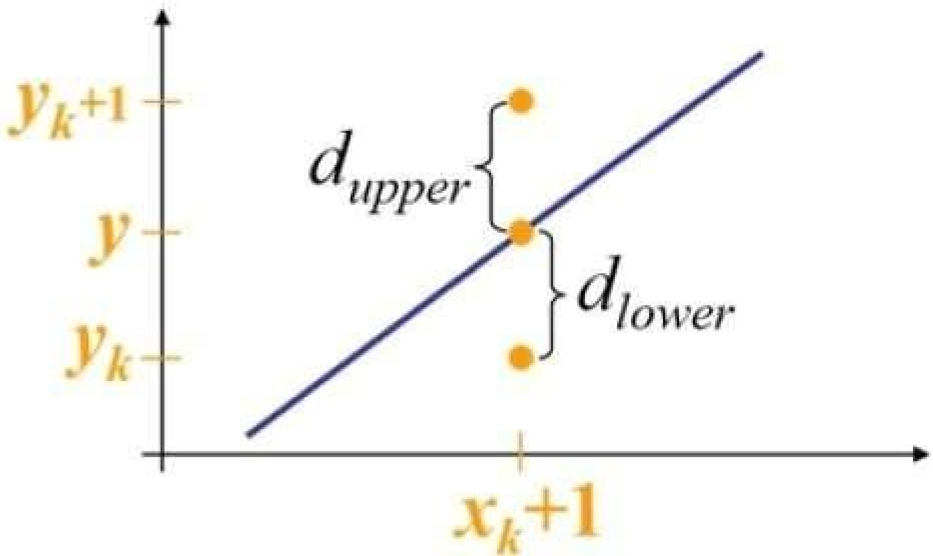
- We need a decision parameter to decide which pixel to choose.



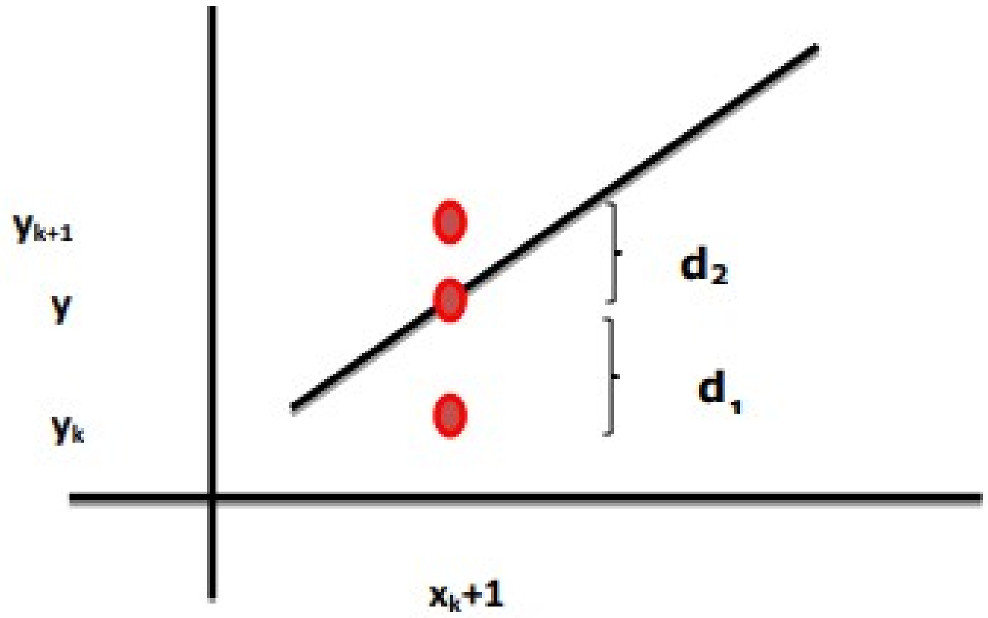
(x k+1,y k+1)
(xk , yk)
(xk+1,yk)



Y
X
Yk+1
y
Yk
(0,0)
Xk
Xk+1



yk+1
y
yk
d upper
d lower
xk+1



- The sign of p_k depends on d_{lower}-d_{upper}.
- If p_k<0(-ve),then we choose the lowerpixel.
- If p_k>0(+ve),then we choose the upperpixel.
- P_k determines which pixel is to be choosen.

Bresenham's Line-Drawing Algorithm for I mI<1

1. Input the twoline endpoints and store the left endpoint in (x_0, y_0)

2. Load (x_0, y_0)into the frame buffer; that is, plot the first point.

3. Calculate constants Δx, Δy, $2\Delta y$, and $2\Delta y$ - $2\Delta x$, and obtain the startingvalue for the decision parameter as

$$p_0 = 2\Delta y - \Delta x$$

4. At each $x_k$along the line, starting at k = 0, perform the following test:

If $P_k < 0$, the next point to plot is $(x_k+1, y_k)$and

$$P_{k+1} = P_k + 2\Delta y$$

Otherwise, the next point to plot is $(x_k + 1, y_k + 1)$ and

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 Δx times.

EXAMPLE

- End points (20,10) and (30,18)
  - $\Delta x = x2 - x1 = 30 - 20 = 10$
  - $\Delta y = y2 - y1 = 18 - 10 = 8$
  - $m = \Delta y / \Delta x = 8/10 = 0.8$

k	P_k	(x_{k+1}, y_{k+1})
0	$6 > 0$	(21,11)
1	$2 > 0$	(22,12)
2	$-2 < 0$	(23,12)
3	$14 > 0$	(24,13)
4	$10 > 0$	(25,14)

k	P_k	(x_{k+1}, y_{k+1})
5	$6 > 0$	(26,15)
6	$2 > 0$	(27,16)
7	$-2 < 0$	(28,16)
8	$14 > 0$	(29,17)
9	$10 > 0$	(30,18)

ADVANTAGES

- Bresenham's algorithm generally faster than DDA
- Bresenham's algorithm uses integer arithmetic
- Constants need to be computed only once
- Reduces complexity
- More accurate than DDA

CIRCLE DRAWING ALGORITHMS

There are two algorithms for generating a circle

- Midpoint circle drawing algorithm
- Bresenham’s circle drawing algorithm

MID POINT CIRCLE DRAWING ALGORITHM

- Given radius 'r' and center (x_c, y_c)
- We first setup our algorithm to calculate circular path coordinates for center (0, 0). And then we will transfer calculated pixel position to center (x_c, y_c) by adding x_c to x and y_c to y.
- Along the circle section from $x = 0$ to $x = y$ in the first quadrant, the slope of the curve varies from 0 to -1 so we can step unit step in positive x direction over this octant and use a decision parameter to determine which of the two possible y position is closer to the circular path.
- Position in the other seven octants are then obtain by symmetry.
- For the decision parameter we use the circle function which is:

$$f_{circle}(x, y) = x^2 + y^2 - r^2$$

To apply the midpoint method, a circle function is defined:

$$f_{circle}(x,y) = x^2 + y^2 - r^2$$

Any point (x, y) on the boundary of the circle with radius r satisfies the equation $f_{circle}(x,y) = 0$.

- If the point is in the interior of the circle, the circle function is negative.
- And if the point is outside the circle, the circle function is positive.

Hence, the relative position of any point(x,y) can be determined by checking the sign of the circle function:

$$f_{circle}(x,y) \begin{cases} < 0, & \text{if } (x,y) \text{ is inside the circle boundary} \\ = 0, & \text{if } (x,y) \text{ is on the circle boundary} \\ > 0, & \text{if } (x,y) \text{ is outside the circle boundary} \end{cases}$$

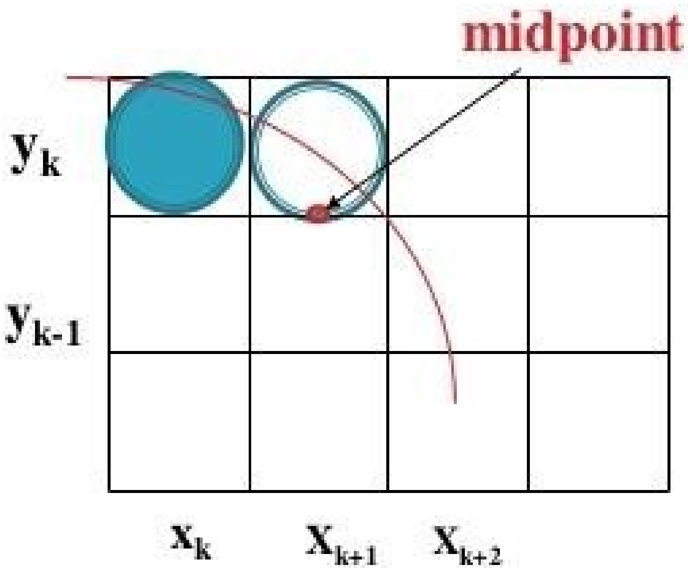
- Assuming we have just plotted the pixel at (x_k, y_k) and next we need to determine whether the pixel at position '$(x_k + 1, y_k)$' or the one at position' $(x_k + 1, y_k - 1)$' is closer to circle boundary.
- So for finding which pixel is more closer using decision parameter evaluated at the midpoint between two candidate pixels as below:

$$p_k = f_{circle}\left(x_k + 1, y_k - \frac{1}{2}\right)$$

$$p_k = (x_k + 1)^2 + \left(y_k - \frac{1}{2}\right)^2 - r^2$$

- If $p_k < 0$ this midpoint is inside the circle and the pixel on the scan line y_k is closer to circle boundary. Otherwise the midpoint is outside or on the boundary and we select the scan line $y_k - 1$.

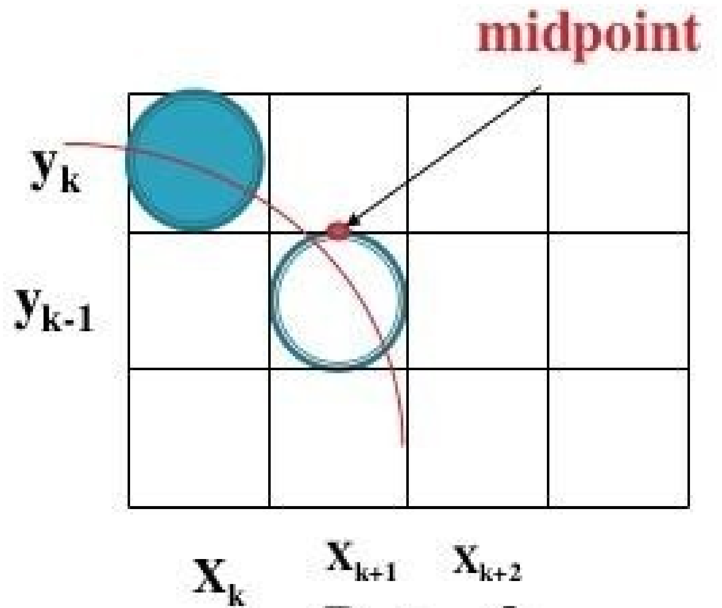
Midpoint Circle Drawing Algorithm



$$P_k < 0$$

$$y_{k+1} = y_k$$

Next pixel = (x_k+1, y_k)



$$P_k >= 0$$

$$y_{k+1} = y_k - 1$$

Next pixel = (x_k+1, y_k-1)

$$p_{k+1} = f_{circle}\left(x_{k+1} + 1, y_{k+1} - \frac{1}{2}\right)$$

$$p_{k+1} = [(x_k + 1) + 1]^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2$$

- Now we can obtain recursive calculation using equation of p_{k+1} and p_k as follow.

$$p_{k+1} - p_k = \left([(x_k + 1) + 1]^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2\right) - \left((x_k + 1)^2 + \left(y_k - \frac{1}{2}\right)^2 - r^2\right)$$

$$p_{k+1} = p_k + 2(x_k + 1) + (y_{k+1}{}^2 - y_k{}^2) - (y_{k+1} - y_k) + 1$$

- In above equation y_{k+1} is either y_k or $y_k - 1$ depending on the sign of the p_k.
- Now we can put $2x_{k+1} = 2x_k + 2$ and when we select $y_{k+1} = y_k - 1$ we can obtain $2y_{k+1} = 2y_k - 2$.

- The initial decision parameter is obtained by evaluating the circle function at the start position $(x_0, y_0) = (0, r)$ as follows.

$$p_0 = f_{circle}\left(0+1, r-\frac{1}{2}\right)$$

$$p_0 = 1^2 + \left(r-\frac{1}{2}\right)^2 - r^2$$

$$p_0 = 1 + r^2 - r + \frac{1}{4} - r^2$$

$$p_0 = \frac{5}{4} - r$$

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 centered on the origin as

$$(x_0, y_0) = (0, r)$$

2. Calculate the initial value of the decision parameter as

$$p_0 = \frac{5}{4} - r$$

3. At each x_k position, starting 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

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

   Otherwise, the next point along the circle is $(x_k + 1, y_k - 1)$ and

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

   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 coordinate values:

$$x = x + x_c, \qquad y = y + y_c$$

6. Repeat steps 3 through 5 until $x \geq y$.

Midpoint Circle Drawing Algorithm

Example:

Given a circle radius = 10, determine the circle octant in the first quadrant from x=0 to x=y.

Solution:

$$p_0 = \frac{5}{4} - r$$

$$= \frac{5}{4} - 10$$

$$= -8.75$$

$$\approx -9$$

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

k	P_k
0	-9
1	-9+2+1=-6
2	-6+4+1=-1
3	-1+6+1=6
4	6+8+1-18=-3
5	-3+10+1=8
6	8+12+1-16=5

k	p_k	p_{k+1}	(x_{k+1}, y_{k+1})
0			(0, 10)
1	-9	-6	(1, 10)
2	-6	-1	(2, 10)
3	-1	6	(3, 10)
4	6	-3	(4, 9)
5	-3	8	(5, 9)
6	8	5	(6, 8)
7	?	?	

Bresenham's circle algorithm

1. Input the radius r of the circle .
2. Initialize the decision parameter p0 = 3 – (2 * r).
3. x0=0; and y0=r;
4. Plot the pixels until x < = y
   - If $p_k < 0$, $p_{k+1} = p_k + (4*x) + 6$

     $x_{k+1} = x_k + 1;$
   - Else, $p_{k+1} = p_k + 4(x_k - y_k) + 10$

     $x_{k+1} = x_k + 1; y_{k+1} = y_k - 1;$
5. Plot(x,y)
6. Determine the symmetry points in other seven octants.
7. Stop.

- The radius of a circle is 8, and center point coordinates are (0, 0). Apply bresenham's circle drawing algorithm to all plot points of the circle.

(xo ,yo) =(0,r) =(0,8)

Now, we will calculate the initial decision parameter **(P_0)**

$\mathbf{P_0 = 3 - 2 \times r}$

$\mathbf{P_0 = 3 - 2 \times 8}$

$\mathbf{P_0 = -13}$

The value of initial parameter $\mathbf{P_0 < 0}$. So, case 1 is satisfied.

Thus,

$\mathbf{x_{k+1} = x_k + 1 = 0 + 1 = 1}$

$\mathbf{y_{k+1} = y_k = 8}$

$\mathbf{P_{k+1} = P_k + 4x_{k+1} + 6 = -13 + (4 \times 1) + 6 = -3}$

- Follow step 3 until we get **x >= y.**

k	p_k	p_{k+1}	(x_{k+1}, y_{k+1})
0			**(0, 8)**
1	**-13**	**-3**	**(1, 8)**
2	**-3**	**11**	**(2, 8)**
3	**11**	**5**	**(3, 7)**
4	**5**	**7**	**(4, 6)**
5	**7**		**(5, 5)**