CST 304- COMPUTER GRAPHICS & IMAGE PROCESSING

SLOT , CREDIT 4

Prepared by : Asha Baby

AP in CSE VJEC

COURSE OBJECTIVES

- The purpose of this course is to make awareness about strong **theoretical relationships between computer graphics and image processing.**

- This course helps the learner to understand **three-dimensional environment representation** in a computer, **transformation of 2D/3D objects**, basic mathematical techniques and **algorithms used to build useful applications**, imaging, and **image processing techniques.**

- The study of computer graphics and image processing develops the ability to create image processing frameworks for **different domains and develops algorithms** for emerging display technologies.

COURSE OUTCOMES

CO#	CO
CO1	Describe the working principles of graphics devices**(Cognitive Knowledge level: Understand)**
CO2	Illustrate line drawing, circle drawing and polygon filling algorithms(Cognitive Knowledge level: Apply)
CO3	Demonstrate geometric representations, transformations on 2D & 3D objects, clipping algorithms and projection algorithms(Cognitive Knowledge level: Apply)
CO4	Summarize visible surface detection methods(Cognitive Knowledge level: Understand)
CO5	Summarize the concepts of digital image representation, processing and demonstrate pixel relationships(Cognitive Knowledge level: Apply)
CO6	Solve image enhancement and segmentation problems using spatial domain techniques(Cognitive Knowledge level: Apply)

Syllabus

Syllabus

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.

Module - 2(Filled Area Primitives and transformations)

Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling. Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.

Module - 3 (Clipping and Projections)

Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman Polygon clipping algorithm. Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Depth buffer algorithm, Scan line algorithm.

Module - 4 (Fundamentals of Digital Image Processing)

Introduction to Image processing and applications. Image as 2D data. Image representation in Gray scale, Binary and Colour images. Fundamental steps in image processing. Components of image processing system. Coordinate conventions. Sampling and quantization. Spatial and Gray Level Resolution. Basic relationship between pixels– neighbourhood, adjacency, connectivity. Fundamentals of spatial domain-convolution operation.

Module - 5 (Image Enhancement in Spatial Domain and Image Segmentation)

Basic gray level transformation functions - Log transformations, Power-Law transformations, Contrast stretching. Histogram equalization. Basics of spatial filtering - Smoothing spatial filter-Linear and nonlinear filters, and Sharpening spatial filters-Gradient and Laplacian.

Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.

Text Book

1. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996
2. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Pearson, 4e, 2017

References

1) William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2001

MODULE I

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.

Basic concepts in Computer Graphics

- **Computer graphics** is an art of drawing pictures on computer screens with the help of programming.
- It involves computations, creation and manipulation of data.

Basically there are two types of computer graphics

- Interactive computer graphics
- Non interactive computer graphics

Interactive Computer Graphics:

- **Interactive computer graphics** work using the concept of two-way communication between computer and users.
- The computer will receive signals from the input device, and the picture is modified accordingly.
- Picture will be changed quickly when we apply command.
- In interactive Computer Graphics user have some controls over the picture, i.e., the user can make any change in the produced image.

- One example of it is the ping-pong game.

Advantages:

- Higher Quality
- More precise results or products
- Greater Productivity
- Significantly enhances our ability to understand data and to perceive trends.

Non-Interactive or Passive Computer Graphics:

- In **non-interactive computer graphics**, the picture is produced on the monitor, and the user does not have any controlled over the image, i.e., the user cannot make any change in the rendered image.

- One example of its Titles shown on T.V.

- Non-interactive Graphics involves only one-way communication between the computer and the user, User can see the produced image, and he cannot make any change in the image.

Application of Computer Graphics

Education and Training

- Computer-generated model of the physical, financial and economic system is often used as educational aids.

- Model of physical systems, physiological system, population trends or equipment can help trainees to understand the operation of the system.

- For some training applications, particular systems are designed. For example Flight Simulator.

- **Flight Simulator:**
- It helps in giving training to the pilots of airplanes.
- These pilots spend much of their training not in a real aircraft but on the ground at the controls of a Flight Simulator.
- **Use in Biology:**
- Molecular biologist can display a picture of molecules and gain insight into their structure with the help of computer graphics.

Computer-Generated Maps:

- Town planners and transportation engineers can use computer-generated maps which display data useful to them in their planning work.

Architect:

- Architect can explore an alternative solution to design problems at an interactive graphics terminal.

- In this way, they can test many more solutions that would not be possible without the computer.

Presentation Graphics:

- Example of presentation Graphics are bar charts, line graphs, pie charts and other displays showing relationships between multiple parameters.

- Presentation Graphics is commonly used to summarize, financial reports, statistical reports, mathematical reports, scientific reports, economic Data for research reports, managerial reports, consumer information bulletins, and other types of reports.

Computer Art:

- Computer Graphics are also used in the field of commercial arts.
- It is used to generate television and advertising commercial.

Entertainment

- Computer graphics are now commonly used in making motion pictures, music videos and television shows.

Visualization:

- It is used for visualization of scientists, engineers, medical personnel, business analysts for the study of a large amount of information.

Educational Software:

- Computer Graphics is used in the development of educational software for making computer-aided instruction.

Printing Technology:

- Computer Graphics is used for printing technology and textile design.

Example of Computer Graphics Packages:

- LOGO
- COREL DRAW
- AUTO CAD
- 3D STUDIO
- CORE
- GKS (Graphics Kernel System)
- PHIGS
- CAM (Computer Graphics Metafile)
- CGI (Computer Graphics Interface)

Pixel

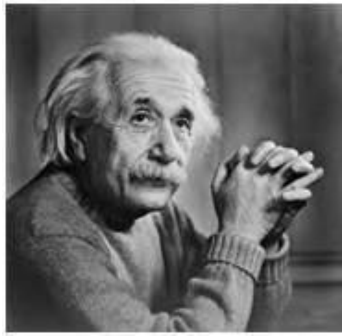
- Pixel is the smallest element of an image.

- Each pixel correspond to any one value.

- In an 8-bit gray scale image, the value of the pixel between 0 and 255.

- The value of a pixel at any point correspond to the intensity of the light photons striking at that point.

- Each pixel store a value proportional to the light intensity at that particular location.



In the first picture, there may be thousands of pixels, that together make up this image.

We will zoom that image to the extent that we are able to see some pixels division.

It is shown in the image below.

Calculation of total number of pixels

- We have define an image as a two dimensional signal or matrix.
- Then in that case the number of **PEL would be equal to the number of rows multiply with number of columns.**
- This can be mathematically represented as below:

Total number of pixels = number of rows (X) number of columns

Gray level

- The value of the pixel at any point denotes the intensity of image at that location, and that is also known as gray level.

Pixel value

- Each pixel can have only one value and each value denotes the intensity of light at that point of the image.
- The value 0 means absence of light. It means that 0 denotes dark, and it further means that when ever a pixel has a value of 0, it means at that point, black color would be formed.
- Have a look at this image matrix

0	0	0
0	0	0
0	0	0

- Total no of pixels = total no. of rows X total no. of columns

  $= 3 \times 3 = 9$ pixels.

- An image would be formed with 9 pixels, and that image would have a dimension of 3 rows and 3 column and most importantly that image would be black.

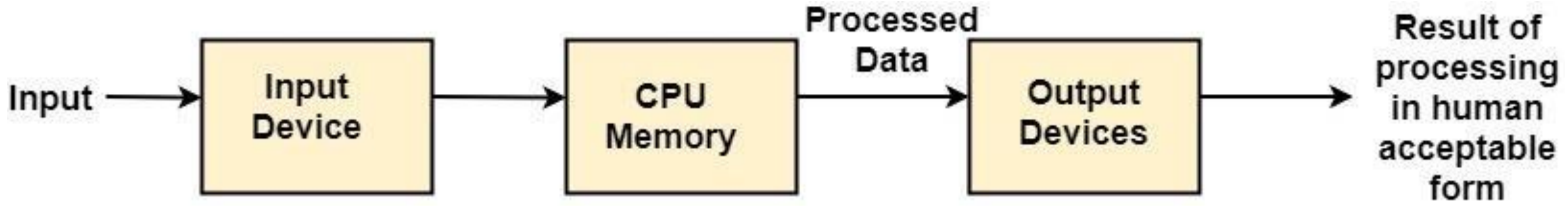
- The resulting image that would be made would be something like this

- **Aspect ratio** is the ratio of the width of an image to the height of the image.
- This ratio is expressed as x:y, and differs in case of different images used in photography, television, computer applications and so on.
- The **resolution** of an image is the total number of pixels displayed on your computer or television screen.
- Generally, the higher the resolution, the higher is the quality of the image.

Aspect ratio	Resolution
Ratio of the width of an image to the height of the image (x:y).	Total number of pixels displayed on your computer or television screen without overlap.
The common aspect ratios used are 4:3,16:9	The common monitor resolutions are 640x480, 800x600 and 1024x768.

Input devices

- The **input devices** are the hardware that is used to transfer input to the computer.
- The data can be in the form of text, graphics, sound, and text.
- **Output device** display data from the memory of the computer.
- Output can be text, numeric data, line, polygon, and other objects.



Output devices

Following are some of the important output devices used in a computer.

Display devices

Graphic Plotter

Printer

Display devices

- The display device is an output device used to represent the information in the form of images (visual form).

- Display systems are mostly called a **video monitor** or **Video display unit (VDU).**

- The purpose of display technology is to simplify information sharing.

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

Display devices

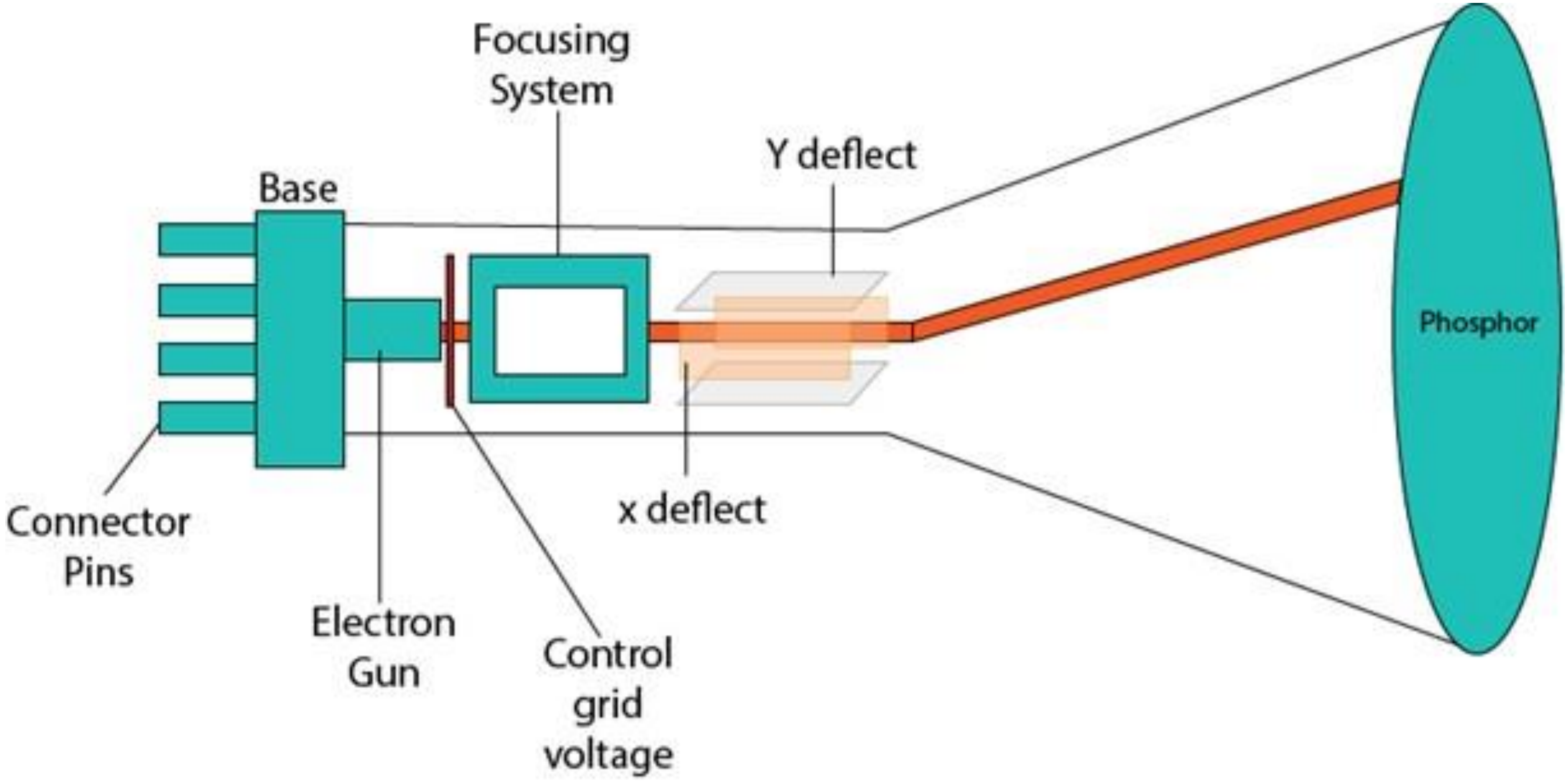
- Cathode-Ray Tube(CRT)
- Color CRT Monitor
- Liquid crystal display(LCD)
- Light Emitting Diode(LED)
- Direct View Storage Tubes(DVST)
- Plasma Display
- 3D Display

Cathode-Ray Tube(CRT)

- The primary output device in a graphics system is a video monitor.
- The operations of the video monitors is based on the standard cathode ray tubes design.
- A CRT is an electronic tube designed to display electrical data.

The basic CRT consists of 4 major components

- Electron gun
- Focusing and accelerating anode
- Horizontal and vertical deflection plates
- Evacuated glass envelop



Cathode Ray Tube

- A beam of electrons (cathode rays), emitted by an electron gun, passes through focusing and deflection systems that direct the beam toward specified positions on the phosphor coated screen.

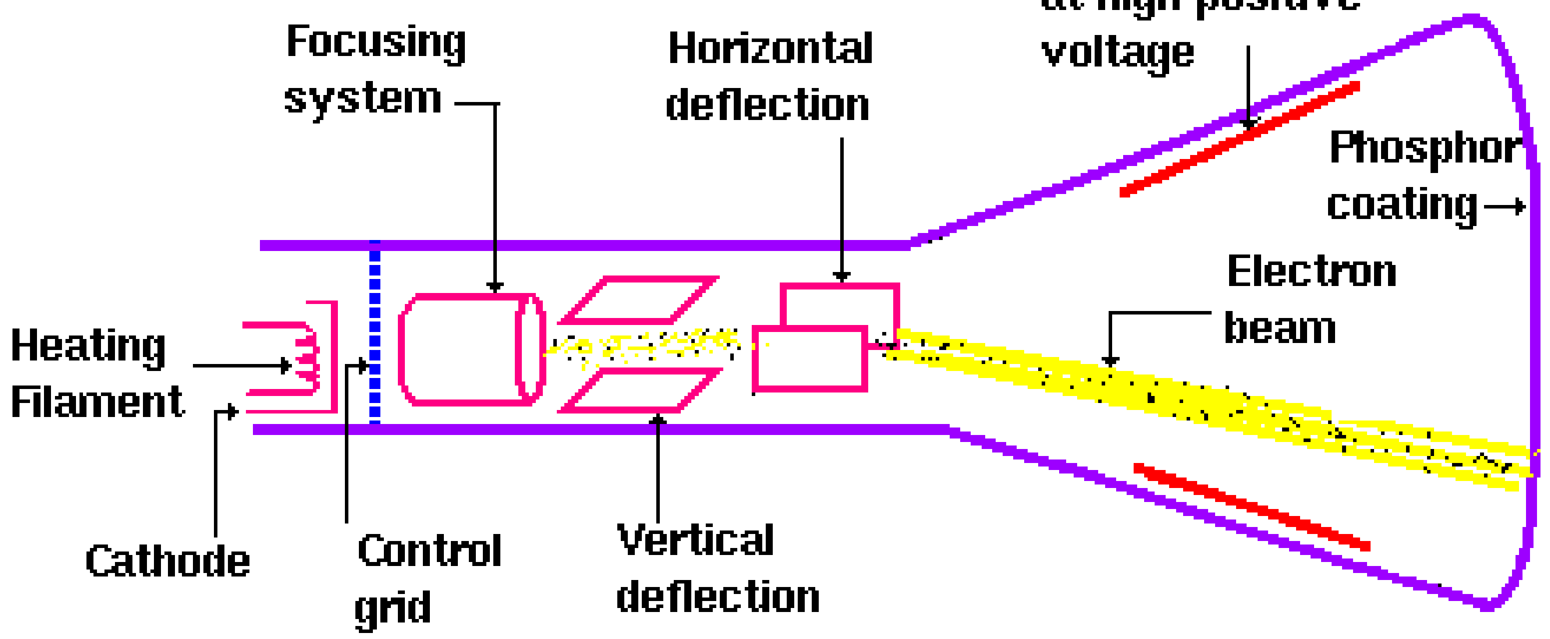
- The phosphor then emits a small spot of light at each position contacted by the electron beam.

- 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**

Cathode Ray Tube (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**, 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 toward the phosphor coating by a high positive voltage.

- The accelerating voltage can be generated with a positively charged metal coating on the inside of the CRT envelope near the phosphor screen, or an accelerating anode can be used.

- Sometimes the electron gun is built to contain the accelerating anode and focusing system within the same unit.

- 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.

- Since the amount of light emitted by the phosphor coating depends on the number of electrons striking the screen, we control the brightness of a display by varying the voltage on the control grid.

- The **focusing system** in a CRT is needed to 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 either electric or magnetic fields. Electrostatic focusing is commonly used in television and computer graphics monitors.

- As with focusing, deflection of the electron beam can be controlled either with electric fields or with magnetic fields.

- Different kinds of phosphors are available for use in a CRT. Besides color, a major difference between phosphors is their persistence, how long they continue to emit light after the CRT beam is removed.

- **Persistence** is defined as the time it takes the emitted light from the screen to decay to one-tenth of its original intensity.

- **Lower persistence** phosphors require higher refresh rates to maintain a picture on the screen without flicker.

- A phosphor with **low persistence** is useful for **animation**, a high-persistence phosphor is useful for displaying highly complex, static pictures.

- Although some phosphors have a persistence greater than 1 second, graphics monitors are usually constructed with a persistence in the range from **10 to 60 microseconds.**

- The maximum number of points that can be displayed without overlap on a CRT is referred to as the **resolution**.

- A more precise definition of resolution is the number of points per centimeter that can be plotted horizontally and vertically, although it is often simply stated as the total number of points in each direction.

- Resolution of a CRT is dependent on the type of phosphor, the intensity to be displayed, and the focusing and deflection systems.

- Typical resolution on high-quality systems is 1280 by 1024, with higher resolutions available on many systems.
- High- resolution systems are often referred to as high-definition systems.
- Another property of video monitors is **aspect ratio**.
- It is the ratio of the width of an image to the height of the image

- (Sometimes aspect ratio is stated in terms of the ratio of horizontal to vertical points.)

- An aspect ratio of 3/4 means that a vertical line plotted with three points has the same length as a horizontal line plotted with four points

There are two ways by which we can display an object on the screen.

Raster scan display and random scan display

Raster scan display

- The most common type of graphics monitor employing a CRT is the raster-scan display, based on television technology.

- In a raster-scan system, the electron beam is swept across the screen, one row at a time from top to bottom.

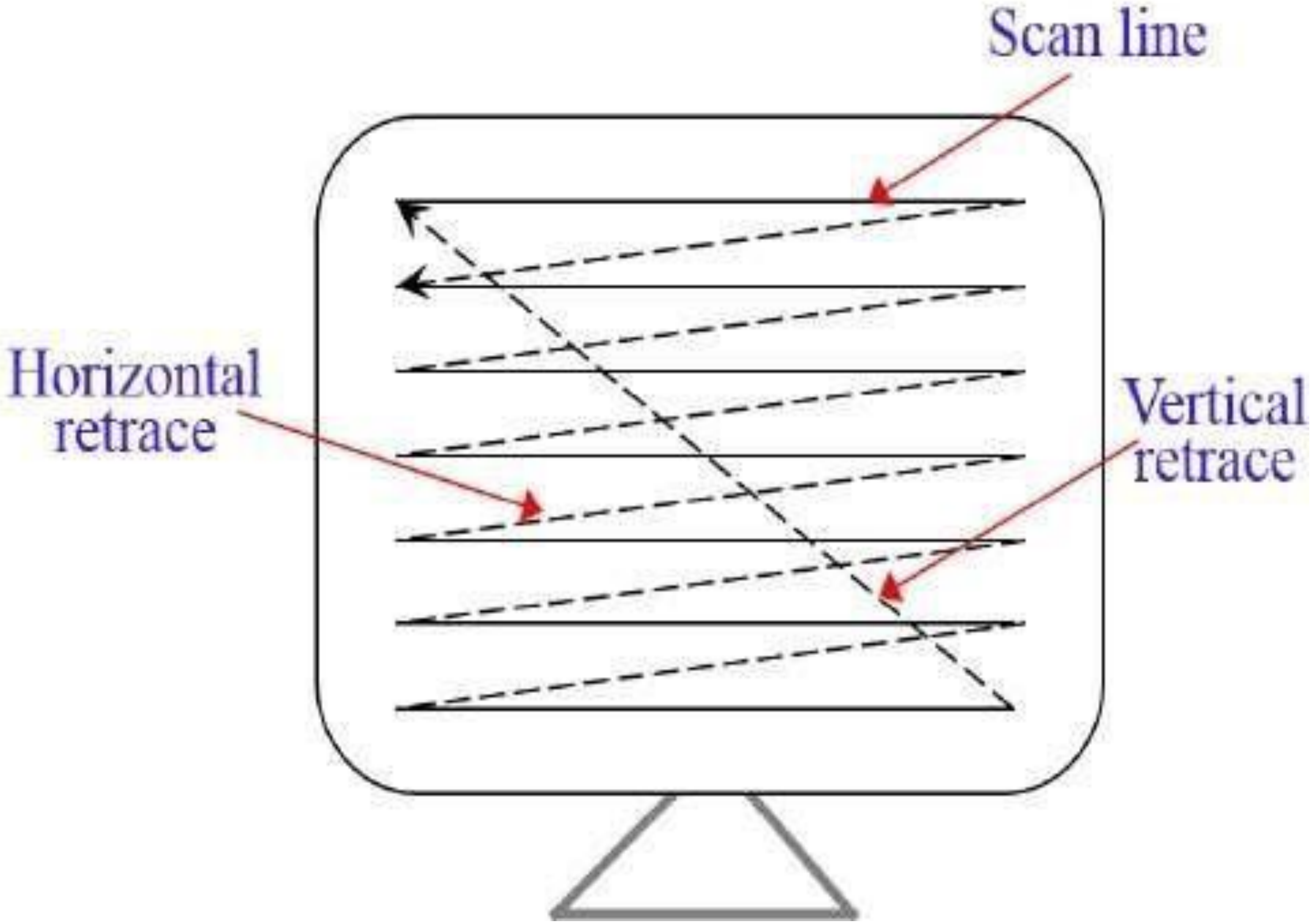
- As the electron beam moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.

- Picture definition is stored in a memory area called the **refresh buffer or frame buffer.**

- This memory area holds the set of intensity values for all the screen points.

- Stored intensity values are then retrieved from the refresh buffer and "painted" on the screen one row (scan line) at a time .

- Each screen point is referred to as a **pixel or pel** (shortened forms of picture element).


Scan line
Horizontal retrace
Vertical retrace

- Raster scan display used in home television sets and printers.

- 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.

- For a bilevel system, a **bit value of 1** indicates that the electron beam is to be turn **on** at that position, and **a value of 0** indicates that the beam intensity is to be **off.**

- **Additional bits** are needed when color and intensity variations can be displayed.

- Up to **24 bits per pixel** are included in high-quality systems, which can require several megabytes of storage for the frame buffer, depending on the resolution of the system.

- A system with 24 bits per pixel and a screen resolution 1024 by 1024 requires **3 megabytes** of storage for the frame buffer.

- On a black-and-white system with one bit per pixel, 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**, although some systems are designed for higher refresh rates.

- Sometimes, refresh rates are described in units of **cycles per second, or Hertz (Hz)**, where a cycle corresponds to one frame.

- Using these units, we would describe a refresh rate of **60 frames per second as simply 60 Hz.**

- 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.

- The return to the left of the screen, after refreshing each scan line, is called the horizontal retrace of the electron beam.

- And at the end of each frame (displayed in 1/80th to 1/60th of a second), the electron beam returns (vertical retrace) to the top left comer of the screen to begin the next frame.

Random scan display

- When operated as random scan display unit, a CRT has the electron beam directed only to the parts of the screen where a picture is to be drawn.

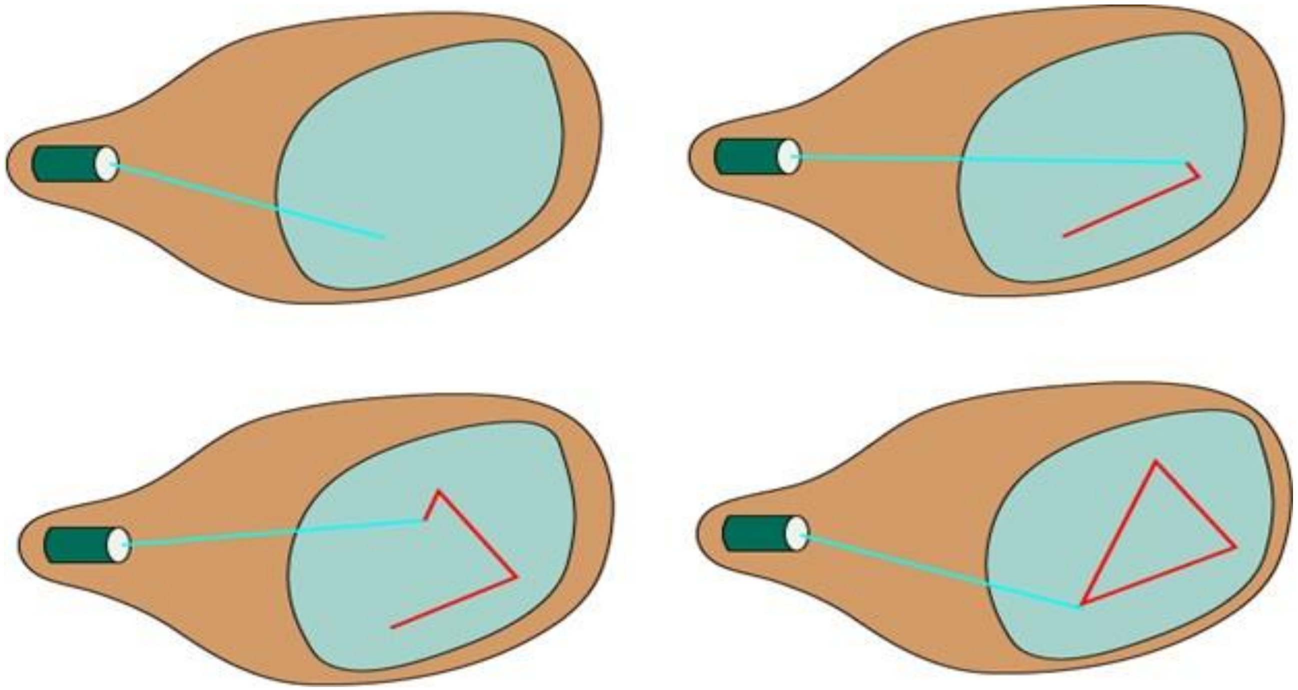
- Random scan monitors draw a picture one line at a time and for this reason are also referred to as **vector displays**.

- A pen plotter operates in a similar way and is an example of a random-scan, hard-copy device.

- Refresh rate on a random scan system depends on the number of lines to be displayed.

- Picture definition is now stored as a set of line drawing commands in an area of memory referred to as the **refresh display file**.

- Sometimes the refresh display file is called the **display list, display program, or simply the refresh buffer**



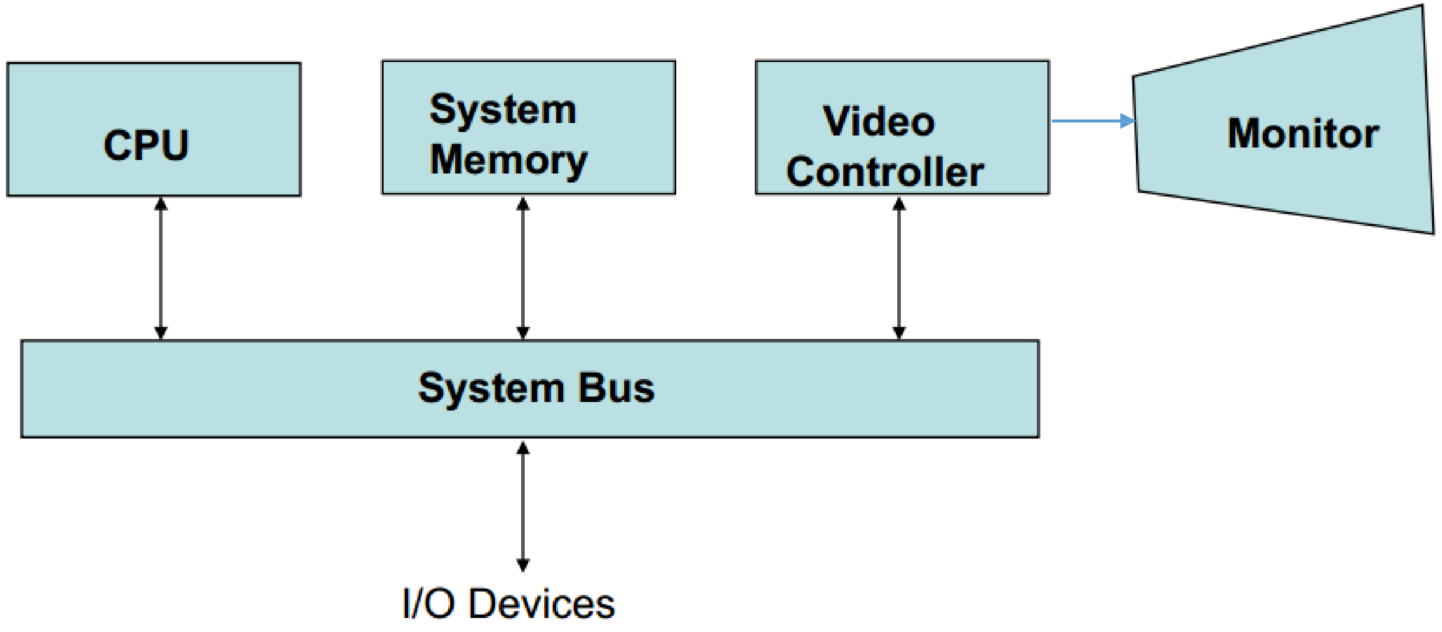
- To display a specified picture, the system cycles through the set of commands in the display file, drawing each component line in turn.
- After all line drawing commands have been processed, the system cycles back to the first line command in the list.
- Random scan displays are designed to draw all the component lines of a picture **30 to 60 times each second.**
- Random scan systems are designed for line drawing applications and **can not display realistic shaded scenes.**

- Since **picture definition** is stored as a set of line drawing instructions and not as a set of intensity values for all screen points, vector displays generally have higher resolution than raster systems.

- Also vector displays produce smooth line drawings because the CRT beam directly follows the line path.

Difference	Raster scan displays	Random scan displays
resolution	It has poor or **less Resolution** because picture definition is stored as a intensity value.	It has **High Resolution** because it stores picture definition as a set of line commands.
Electron beam	**Electron Beam** is directed from top to bottom and one row at a time on screen, but electron beam is directed to whole screen.	**Electron Beam** is directed to only that part of screen where picture is required to be drawn, one line at a time so also called **Vector Display**.
Cost	It is **less expensive**	It is **Costlier** than Raster Scan System.
Refresh rate	Refresh rate is **60 to 80 frame per second**.	Refresh Rate depends on the number of lines to be displayed i.e **30 to 60/sec**
Picture definition	It Stores **picture definition** in **Refresh Buffer** also called **Frame Buffer**.	It Stores picture definition as a set of **line commands** called **Refresh Display File**.
Line drawing	**Zig – Zag line** is produced because plotted value are discrete.	**Smooth line** is produced because directly the line path is followed by electron beam
Realism in display	It contains shadow, advance shading and hidden surface technique so gives the **realistic display** of scenes.	It does not contain shadow and hidden surface technique so **it can not give realistic display of scenes.**
Image drawing	It uses **Pixels** along scan lines for drawing an image.	It is designed for **line drawing applications** and uses various mathematical function to draw.

RASTER SCAN SYSTEMS



Architecture of Simple Raster graphics system

- Interactive raster graphics systems typically employ several processing units.

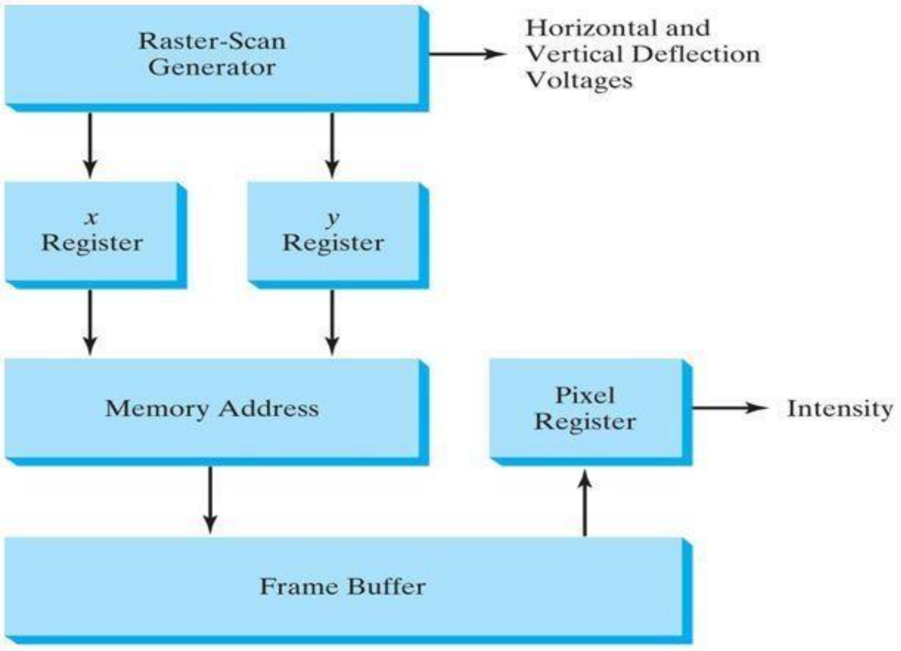
- In addition to the central processing unit, or CPU, a special-purpose processor, called the **video controller or display controller**, is used to control the operation of the display device.

- The frame buffer can be anywhere in the **system memory**, and the video controller accesses the frame buffer to refresh the screen.

Video Controller

Operation of a video controller

Figure 2-19 Basic video-controller refresh operations.



- 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 then 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}

Basic refresh operations

- **Two registers** are used to store the coordinates of the screen pixels.

- Initially, the **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.

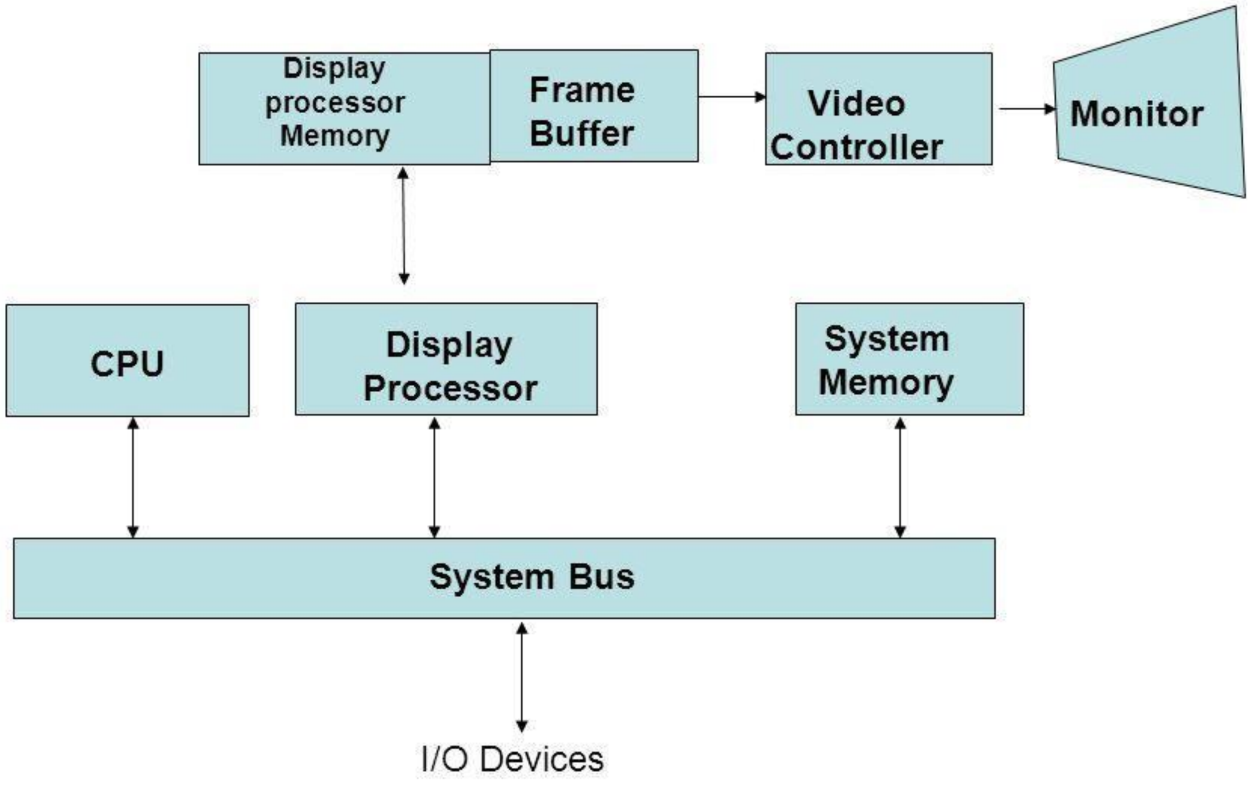
- After cycling through all pixels along the bottom scan line (y = 0), the video controller resets the registers to the first pixel position on the top scan line and the refresh process starts over.

- Since the screen must be refreshed at the rate of 60 frames per second, the simple procedure cannot be accommodated by typical RAM chips.

- The cycle time is too slow. 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.



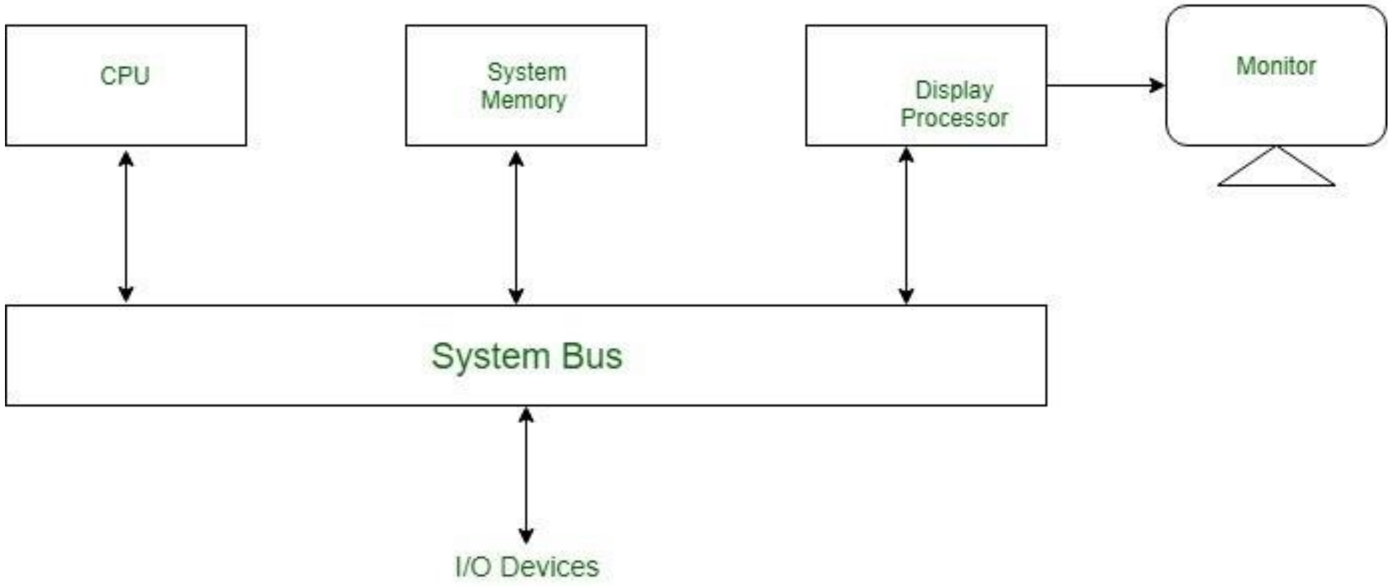
Architecture of Raster graphics system with a display processor

- In addition to the CPU ,the raster graphics system is also contains a separate display processor, referred to as **graphics controller and 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.

RANDOM SCAN SYSTEMS



- It is also known as **vector system.**
- 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.

Suppose you have a raster system designed using an 8 inches × 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? (4)

Answer

- Here, resolution = **8 inch X 10 inch**
- First, we convert it in pixel then Now resolution = 8 X 100 by 10 X 100 pixel = **800**
- **X 1000 pixel**
- 1 pixel can store 6 bits
- So, frame buffer size required = $\frac{800 \times 1000 \times 6 \text{ bits}}{8}$ = 600,000 bytes = **6× 10^5 bytes**.

Color CRT Monitors

- A CRT monitor displays color pictures by using a combination of phosphors that emit different colored light.

- By combining the emitted light from the different phosphors, a range of colors can be generated.

- The two basic techniques for producing color 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 onto 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 colors, orange and yellow.

- The screen color at any point, is controlled by the **beam acceleration voltage.**

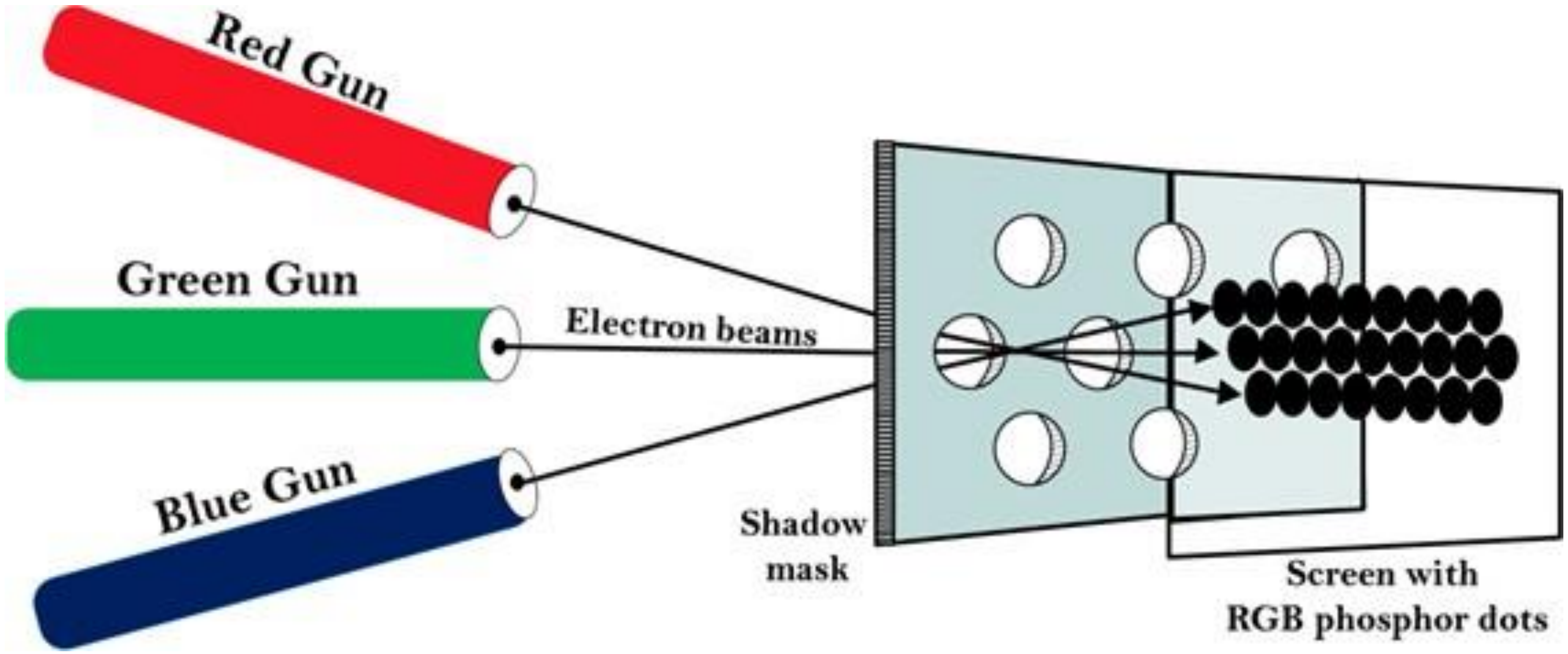
- Beam penetration has been an **inexpensive** way to produce color in random scan monitors.

- But only **four colors** are possible, and the **quality of pictures is not as good** as with other methods.

Shadow-mask methods

- 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 the 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;

- 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 color.

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

POINTS AND LINES

- Line drawing is accomplished by calculating intermediate positions along the line path between two specified endpoint positions.

- The Cartesian slope-intercept equation for a straight line is

  $y=m.x+b$ → equation 1

  m→ slope of the line

  b→ y intercept

- Given that , the two endpoints of a line segment are specified at positions (x_1, y_1) and (x_2, y_2)

  $m = y_2 - y_1 / x_2 - x_1$ → equation 2, $m=\Delta y/\Delta x$ → equation 3

  $b = y_1 - m.x_1$ → equation 4

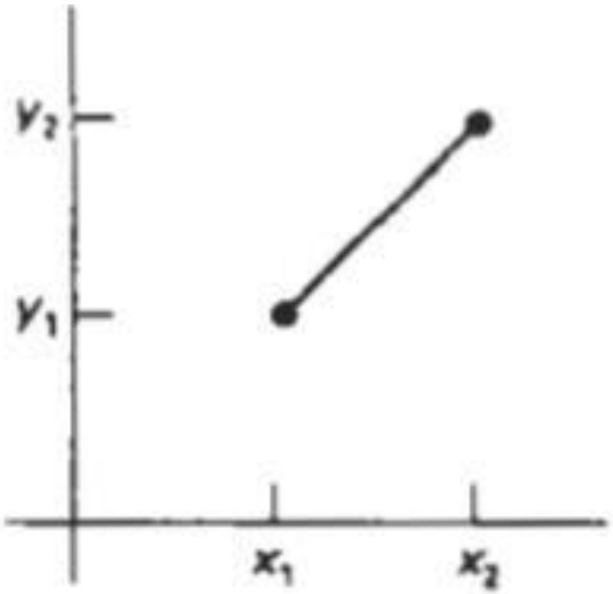


Figure 3-3
Line path between endpoint positions (x_1, y_1) and (x_2, y_2).

Digital Differential Analyzer(DDA)

- Algorithms for displaying straight lines are based on the line equation 1,2 and 3.
- The digital differential analyzer (DDA) is a scan conversion line algorithm based on calculating either Δy or Δx.
- **We sample the line at unit intervals in one coordinate and determine corresponding integer values nearest the line path for the other coordinate.**

Consider first a line with positive slope (the lines are processed from left to right)

- If the slope is less than or equal to 1(m<=1),

  we sample at unit x intervals (Δx = 1) and compute each successive y value as

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

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

y values must be rounded to the nearest integer.

- Subscript k takes integer values starting from 1, for the first point, and increases by 1 until the final endpoint is reached. Since m can be any real number between 0 and 1, the calculated y values must be rounded to the nearest integer

- *(The lines are processed from right to left)*

  $\Delta x = -1$ and

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

For lines with a positive slope greater than 1(m>1)

we reverse the roles of x and y. That is, we sample at unit y intervals ($\Delta y = 1$) and calculate each succeeding x value as

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

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

- If this processing is reversed, so that the starting endpoint is at the right, then either we have $\Delta y = -1$ and

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

Step1 : Start

Step 2: Enter starting and ending points of a line (x_1 , y_1) and (x_2 ,y_2) respectively.

Step 3: Calculate dx = (x_2-x_1) , dy =(y_2-y_1) declare the variables step, k , x_{inc} , y_{inc} and initialize x= x_1, y= y_1 , plot(x, y)

Step 4:Check abs (dx) > abs (dy) , if it is true,

calculate step = abs (dx)

else step = abs (dy)

Step 5: x_{inc}=dx/step

y_{inc} =dy/step

assign x = x_1

assign y = y_1

Step 6:SetPixel (x, y)

Step 7:for(k=0;k<step;k++)

x=x+x_{inc}

y=y+y_{inc}

plot the points (x, y) until k=step, plot the nearest integer value.

Step 8: stop

Sample questions

- Using DDA algorithm plot the line with end points **(20,10)** and (30,18)
- Scan convert the line segment with end points (30,20) and (15,10) using DDA line drawing algorithm . (University qstn 3 M)

- Scan convert the line segment with end points **(20,10)** and (30,18) using DDA.

- X1=20 y1=10 x2=30 y2=18
- dx= 10 dy= 8
- X= 20 y=10
- Abs(dx)>abs(dy) step = 10
- Xinc= 10/10 =1
- Yinc= 8/10=.8
- For k=0 k<10 k++

k	(x,y)
0	(21,10.8)
1	(22,11.6)
2	(23,12.4)
3	(24,13.2)
4	(25,14)
5	(26,14.8)
6	(27,15.6)
7	(28,16.4)
8	(29,17.2)
9	(30,18)

- Scan convert the line segment with end points (30,20) and (15,10) using DDA line drawing algorithm . (University question 4 M)

x1= 30 , y1= 20 x2= 15, y2=10

dx=-15 dy=-10

abs(dx) > abs(dy)

abs(-15) > abs(-10)

Step=15

xinc= -15/15 = -1

yinc= -10/15 = -0.66

k	(x,y)
0	(29,19.34)
1	(28,18.68)
2	(27,18,02)
3	(26,17.36)
4	(25,16.7)
5	(24,16.04)
6	(23,15.38)
7	(22,14.72)
8	(21,14.06)
9	(20,13.4)
10	(19,12.74)
11	(18,12.08)
12	(17,11.42)
13	(16,10.76)
14	(15,10.1)

Advantage:

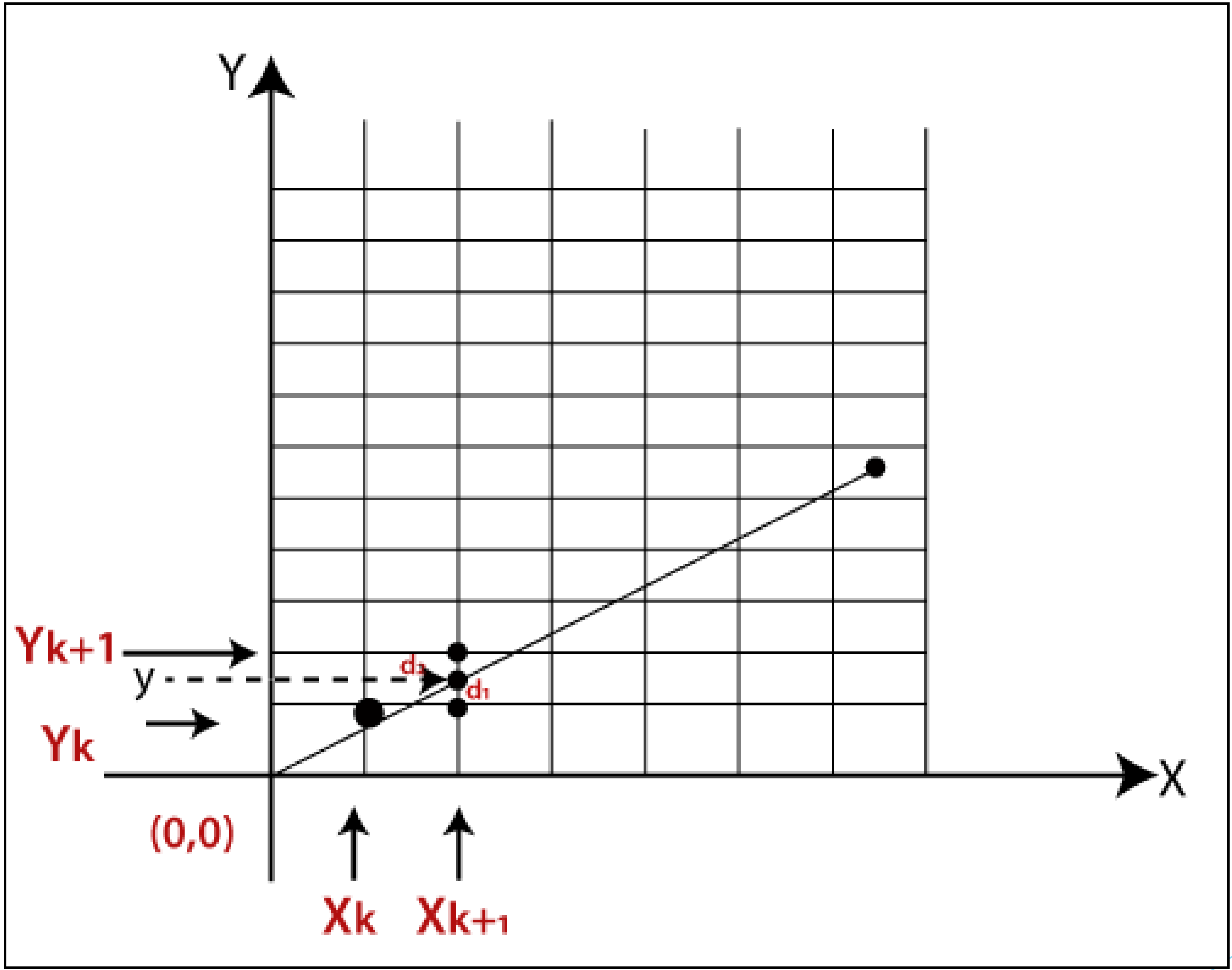
- It is a faster method than method of using direct use of line equation y=m.x+b.
- It eliminates the multiplication by making use of raster characteristics , so that appropriate increments are applied in the x or y direction to step to pixel positions along the line path.
- It is an easy method because each step involves just two additions.

Disadvantage:

- It involves floating point additions rounding off is done. Accumulations of round off error cause accumulation of error.
- Rounding off operations and floating point operations consumes a lot of time.
- It is more suitable for generating line using the software. But it is less suited for hardware implementation.

Bresenham's Line Algorithm

- This algorithm is used for scan converting a line.
- It was developed by Bresenham.
- It is an efficient method because it involves only integer addition, subtractions, and multiplication operations.
- These operations can be performed very rapidly so lines can be generated quickly.


Y
X
Yk+1
y
d2
d1
Yk
(0,0)
Xk
Xk+1

- *Consider the scan conversion process for lines with positive slopes less than 1*
- Starting from the left end- point (x_0 , y_0) of a given line, we step to each successive column (x position) and plot the pixel whose scan line y value is closest to the line path.
- Assuming we have determined that the pixel at (x_k , y_k) is to be displayed, we next need to decide which pixel to plot in column x_{k+1}. Two choices are the pixels at positions (x_k+1, y_k) and (x_k+1, y_k+1).
- We know that , y=m.x+b

$$y=m(x_k+1)+b$$

$$\text{then } d_1 = y-y_k$$

$$= m(x_k+1)+b - y_k$$

$$d_2 = (y_k+1)-y$$

$$= (y_k+1)-(m(x_k+1)+b)$$

The difference between these two separations is ,

$$d_1 - d_2 = m(x_k+1)+b - y_k \quad - (y_k+1)-(m(x_k+1)+b)$$

$$= 2m(x_k+1)-2 y_k +2b-1 \rightarrow \text{equation 1}$$

$$P_K = \Delta x\,(d_1 - d_2)$$

$$= \Delta x \left[2m(x_K+1) - 2y_K + 2b - 1 \right]$$

where $m = \Delta y / \Delta x$

$$P_K = \cancel{\Delta x} \left[2 \times \Delta y / \cancel{\Delta x} \left[x_K + 1 \right] - 2y_K + 2b - 1 \right]$$

$$= 2\Delta y (x_K+1) - \Delta x\, 2 y_K + \Delta x \cdot 2b - \Delta x$$

$$= 2\Delta y x_K + 2\Delta y - 2\Delta x y_K + 2\Delta x b - \Delta x$$

$$= 2\Delta y x_K + 2\Delta y - 2\Delta x y_K + \Delta x (2b-1)$$

$$= 2\Delta y x_K - 2\Delta x y_K + \underline{2\Delta y + \Delta x (2b-1)}$$

$$\boxed{P_K = 2\Delta y x_K - 2\Delta x y_K + c}$$

If the pixel y_K is closer to the line path than the pixel at y_K+1 $(d_1 < d_2)$, P_K is negative. then plot lower pixel.

At step $K+1$, the decision parameter is evaluated from,

$$P_{K+1} = 2\Delta y\, x_{K+1} - 2\Delta x\, y_{K+1} + c$$

$$P_{K+1} - P_K = 2\Delta y\, x_{K+1} - 2\Delta x\, y_{K+1} + c - \left[2\Delta y\, x_K - 2\Delta x\, y_K + c\right]$$

$$= 2\Delta y\,(x_{K+1} - x_K) - 2\Delta x\,(y_{K+1} - y_K)$$

Here $x_{K+1} = x_K + 1$

$$P_{K+1} = P_K + 2\Delta y\,(\cancel{x_K} + 1 - \cancel{x_K}) - 2\Delta x\,(y_{K+1} - y_K)$$

$$\boxed{P_{K+1} = P_K + 2\Delta y - 2\Delta x\,(y_{K+1} - y_K)}$$

where $(y_{K+1} - y_K)$ is either 0 or 1.

Initial decision parameter P_0

$$P_K = 2\Delta y\, x_K - 2\Delta x\, y_K + 2\Delta y + \Delta x\,(2b - 1)$$

$$y = mx + b \quad , \quad b = y - mx.$$

$$P_K = 2\Delta y\, x_K - 2\Delta x\, y_K + 2\Delta y + \Delta x\,(2(y - mx) - 1)$$

$$= 2\Delta y\, x_K - 2\Delta x\, y_K + 2\Delta y + \Delta x\,(2y - 2mx - 1)$$

$$= 2\Delta y\, x_K - 2\Delta x\, y_K + 2\Delta y + 2\Delta x\, y - 2 \cdot \frac{dy}{dx} \cdot dx \cdot x - \Delta x$$

$$= 2\Delta y\, x_K - 2\Delta x\, y_K + 2\Delta y + 2\Delta x\, y - 2dy \cdot x - \Delta x.$$

we know that, initial point is (x_k, y_k) so replace (x, y) with (x_k, y_k)

$$P_k = 2\Delta y x_k - 2\Delta x y_k + 2\Delta y + 2\Delta x y - 2\Delta y x - \Delta x$$

$$= 2\Delta y \cancel{x_k} - 2\Delta x \cancel{y_k} + 2\Delta y + \cancel{2\Delta x y_k} - 2\Delta y . x_k - \Delta x$$

$$\boxed{P_k = 2\Delta y - \Delta x}$$

$$\boxed{P_0 = 2\Delta y - \Delta x}$$

$P_k < 0$

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

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

$P_k \geq 0$

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

$y_{k+1} = y_k$

- If the pixel at y_k is closer to the line path than the pixel at y_k +1(that is, $d_1 < d_2$,), then decision parameter p_k is negative.
- In that case, we plot the lower pixel otherwise, we plot the upper pixel.

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

1. Input the two line 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 starting value 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.

- ***<u>Consider the scan conversion process for lines with positive slopes greater than 1</u>***
- For a line with positive slope greater than 1, we interchange the roles of the x and y directions.
- That is, we step along the y direction in unit steps and calculate successive x values nearest the line path.

- Using bresenham’s line drawing algorithm plot the line with the endpoints (20,10) and (30,18).
- **Calculate the points between the starting coordinates (9, 18) and ending coordinates (14, 22) using bresenham’s line drawing algorithm .**
- Consider the line from (5, 5) to (13, 9). Use the Bresenham's algorithm to rasterize the line.

- Calculate the points between the starting coordinates (9, 18) and ending coordinates (14, 22) using bresenham's line drawing algorithm.
- m=dy/dx dy=4 dx= 5
- m=4/5 = 0.8
- 2dy = 8 , 2dx=10
- P0=2dy-dx   8-5= 3
- p1=p0+2dy-2dx = 3+8-10=1
- P2=p1+2dy-2dx = 1+-2=-1
- P3= p2+2dy = -1+8=7
- P4= p3+8-10 = 7+-2=5

k	pk	(xk+1.yk+1)
0	3	(10,19)
1	1	(11,20)
2	-1	(12,20)
3	7	(13,21)
4	5	(14,22)

- Using bresenham's line drawing algorithm plot the line with the endpoints (20,10) and (30,18).
- calculate m= dy/dx

dy=8 dx=10

m=8/10

=.8

2dy= 16 , 2dx=20

k	p_k	(x_{k+1},y_{k+1})
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)

Advantages of Bresenham Line Drawing Algorithm

- It is easy to implement.
- It is fast and incremental.
- It executes fast but less faster than DDAAlgorithm.
- The points generated by this algorithm are more accurate than DDAAlgorithm.
- It uses fixed points only.

Disadvantages of Bresenham Line Drawing Algorithm

- Though it improves the accuracy of generated points but still the resulted line is not smooth.
- This algorithm is for the basic line drawing.

Circle generating algorithm

Properties of Circles

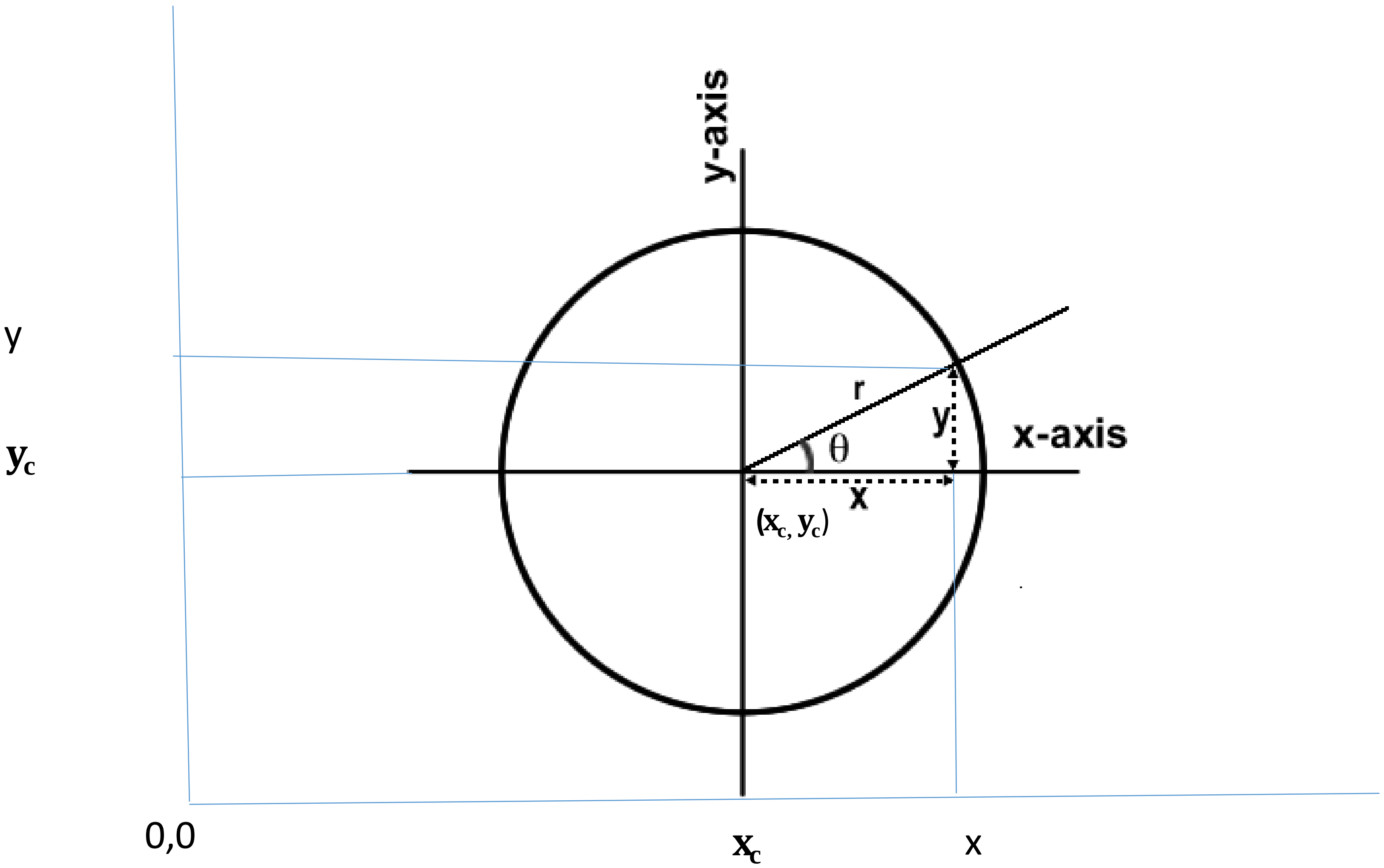
- A circle is defined as the set of points that are all at a given distance r from a center position (x_c,y_c).
- This distance relationship is expressed by the Pythagorean theorem in Cartesian coordinates as,

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

We can use this equation to calculate the positions of points on a circle by stepping along the x axis in unit steps and calculating the corresponding y values at each po sition as ,

$$y = y_c \pm \sqrt{r^2 - (x - x_c)^2}$$

But this is not the best method for getting a circle.


y-axis
y
y_c
r
y
θ
x-axis
x
(x_c, y_c)
0,0
x_c
x

- One problem with this approach is that it involves considerable **computation at each step.**
- Another method is to calculate the points along the circular boundary using **polar coordinates r and θ.**
- Expressing the circle equation in parametric polar form yields the pair of equation.

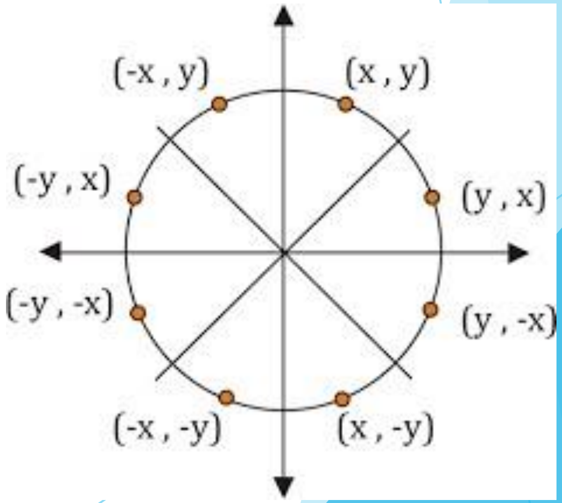
  $x=x_c+r\cos\theta$

  $y=y_c+r\sin\theta$
- This equation require more **computing time**, because it involves multiplication and square root calculation while the parametric equation contains multiplication and trigonometric calculation.
- So we have 2 circle generating algorithm Midpoint circle algorithm, Bresenham's algorithm.

Midpoint circle drawing algorithm

Principle of the midpoint

- **Computation can be reduced** by considering the **symmetry of circles.**
- The shape of the circle is similar in each quadrant.
- Circle sections in adjacent octants within one quadrant are symmetric with respect to the 45 degree line dividing the two octants.



- Taking advantage of the circle symmetry we can generate all pixel positions around a circle by calculating only the points within the sector from x=0 to x=y.

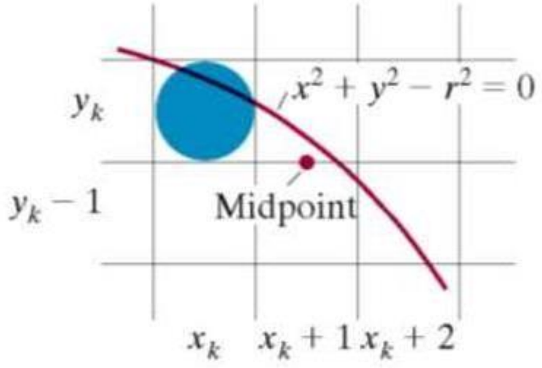


Figure 3-19

Midpoint between candidate pixels at sampling position $x_k + 1$ along a circular path.

- To apply the midpoint method, we define a circle function:

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

- The relative position of any point (x,y) can be determined by checking the sign of the circle function.
- $f_{circle}(x,y)= \{ <0$, 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

- The circle function is the decision parameter in the midpoint algorithm, and we can set up incremental calculations for this function as we did in the line algorithm.
- Assuming we have just plotted the pixel at $(\mathbf{x_k}, \mathbf{y_k})$, we next need to determine whether the pixel at position $(\mathbf{x_k+1}, \mathbf{y_k})$ or the one at position $(\mathbf{x_k+1}, \mathbf{y_k-1})$ is closer to the circle.
- Our decision parameter is the circle function evaluated at the midpoint between these two pixels.

- Find the values from one octant of a circle centered at (0,0), then find the remaining octants by symmetry, then translate to (x_c,y_c).
- The circle function is the decision parameter.
- Calculate the circle function for the midpoint between two pixels.
- If pk<0, midpoint is inside the circle and yk is closer , we select yk
- If pk>=0, midpoint is outside or on the circle and yk-1 is closer, we select yk-1

Assuming we have just plotted the pixel at (x_k, y_k), we next 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 the Circle.

→ Our decision parameter is the Circle function evaluated at the midpoint between these two pixels.

$$\text{midpoint} = (x_k+1, y_k) \text{ and } (x_k+1, y_k-1)$$

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

$$= \left((x_k+1), y_k - 1/2\right)$$

$$P_k = f_{circle}\left[x_k+1, y_k - 1/2\right)$$

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

If $p_k < 0$ this midpoint is inside the circle and the pixel on scan line y_k is closer to the circle boundary.

otherwise the midpoint position is outside or on the circle boundary and we select the pixel on scanline $(y_k - 1)$

Successive decision parameters are obtained using incremental calculations

ie $p_{k+1} = f_{circle}(x_{k+1} + 1, y_{k+1} - 1/2)$, $\boxed{\text{where } x_{k+1} = x_k + 1}$

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

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

$- [(x_k + 1)^2 + (y_k - 1/2)^2 - r^2]$

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

$- [(x_k + 1)^2 + y_k^2 - y_k + 1/4 - r^2]$

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

$- \cancel{(x_k + 1)^2} - y_k^2 + y_k - \cancel{1/4} + \cancel{r^2}]$

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

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

44

where y_{k+1} is either y_k or y_{k-1} depending on the sign of p_k.

$p_k < 0$, $\boxed{(x_k+1, y_k)}$

$$p_{k+1} = p_k + 2(x_k+1) + 1$$

$$= p_k + \underline{2x_k + 2} + 1$$

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

$p_k > 0$, $\boxed{(x_k+1, y_k-1)}$

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

$+(2-2y_k)$
$(2y_k - 2)$
$= 2y_{k+1}$

where $2x_{k+1} = 2x_k + 2$ and $2y_{k+1} = 2y_k - 2$

→ The initial decision parameter is obtained by evaluating the circle function at the start position $(x_0, y_0) = \underline{(0, r)}$

$$p_k = f_{circle}(x_k+1, y_k - 1/2)$$

$$p_0 = f_{circle}(1, r - 1/2)$$

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

$$= 1 + \cancel{r^2} - r + 1/4 - \cancel{r^2}$$

$$= 5/4 - r$$

If the radius r is specified as an integer,

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 value of the decision parameter as $p_0 = 5/4 - r$ **(take this value as 1-r)**

3.At each x_k, from k=0, perform the following test:

if $p_k < 0$, next point to plot along the circle centered on (0,0) is $(x_k + 1, y_k)$ and $p_{k+1} = p_k + 2x_{k+1} + 1$

otherwise, next point to plot 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 at (x_c, y_c) and plot the coordinate values: $x = x + x_c$, $y = y + y_c$

6.Repeat steps 3 through 5 until x >= y.

Sample questions

- The center coordinates are **(0, 0),** and the radius of the circle is **10**. Find all points of the circle by using the midpoint circle drawing algorithm?
- Given the center point coordinates (4, 4) and radius as 10, generate all the points to form a circle.
- Use mid-point circle drawing algorithm to plot a circle whose radius =20 units and center at (50,30) (5M).

The center coordinates are (0, 0), and the radius of the circle is 10. Find all points of the circle by using the midpoint circle drawing algorithm?

- Initial point =(0,10)
- p0=(1-r) (1-10) = -9 next point(1,10) p1= -9+2+1=-6
- Next point(2,10) p2=-6+4+1=-1
- Next point(3,10) p3=-1+6+1= 6
- Next point (4,9) p4=6+8+1-18=-3
- Next point (5,9) p5= -3+10+1=8
- Next point(6,8) p6=8+12+1-16=5

k	P_k	(x_{k+1}, y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0	-9	(1,10)	2	20
1	-6	(2,10)	4	20
2	-1	(3,10)	6	20
3	6	(4,9)	8	18
4	-3	(5,9)	10	18
5	8	(6,8)	12	16
6	5	(7,7)	14	14

Q1(x,y)	Q2(-x,y)	Q3(-x,-y)	Q4(x,-y)
(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)

Q1(x,y)	Q2(-x,y)	Q3(-x,-y)	Q4(x,-y)
(y,x)	(y,-x)	(-y,-x)	(-y,x)

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

- ***We first calculate the points assuming the center coordinates is (0, 0).***
- ***At the end, we translate the circle.***
- Problem same as previous
- Already calculated (x,y) values for all quadrant. Now, we find the values of X_{plot} and Y_{plot}
- $X_{plot} = X_c + X$
- $Y_{plot} = Y_c + Y$

Circle center (4,4)

k	P_k	(x_{k+1}, y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0	-9	(1,10)	2	20
1	-6	(2,10)	4	20
2	-1	(3,10)	6	20
3	6	(4,9)	8	18
4	-3	(5,9)	10	18
5	8	(6,8)	12	16
6	5	(7,7)	14	14

k	P_k	(x_{k+1}, y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0	-9	(5,14)	2	20
1	-6	(6,14)	4	20
2	-1	(7,14)	6	20
3	6	(8,13)	8	18
4	-3	(9,13)	10	18
5	8	(10,12)	12	16
6	5	(11,11)	14	14

Advantages

- It is a powerful and efficient algorithm.
- The entire algorithm is based on the simple equation of circle $X^2 + Y^2 = R^2$.
- It is easy to implement from the programmer's perspective.
- This algorithm is used to generate curves on raster displays.

Disadvantages

- The accuracy of the generating points is an issue in this algorithm.
- The circle generated by this algorithm is not smooth.
- This algorithm is time consuming

Bresenhams 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 value of the decision parameter as $p_0 = 3-2r$

3.At each x_k, from k=0, perform the following test:

if $p_k<0$, next point to plot along the circle centered on (0,0) is $(x_k +1, y_k)$ and $p_k +1 = p_k + 4 x_k + 6$

otherwise, next point to plot is $(x_k + 1, y_k - 1)$ and $p_k +1= p_k + 4(x_k – y_k)+10$

4.Determine symmetry points in the other seven octants.

5.Move each calculated pixel position (x, y) onto the circular path centered at (x_c, y_c) and plot the coordinate values: $x = x + x_c$, $y = y + y_c$

6.Repeat steps 3 through 5 until x >= y.

The center coordinates are (0, 0), and the radius of the circle is 10. Find all points of the circle by using the midpoint circle drawing algorithm?

k	p_k	(x_{k+1}, y_{k+1})
0	-17	(1,10)
1	-11	(2,10)
2	-1	(3,10)
3	13	(4,9)
4	-5	(5,9)
5	17	(6,8)
6	11	(7,7)