SRM Institute of Science and Technology, Chennai

21CSE255T –Computer Graphics and Animation
Unit-I

Unit I- Syllabus

Overview of Computer Graphics, Computer Graphics Application and Software - Video Display devices - Raster scan systems - Random Scan systems - Graphics Monitors and Workstations - Input Devices - HardCopy Devices - Graphics Software - Output Primitives - Points and Lines - Line-Drawing Algorithms - Circle-Generating Algorithms - Ellipse-Generating Algorithms - Parallel Curve Algorithms - Curve Functions - Pixel Addressing - Filled-Area Primitives - Scan-Line Polygon Fill Algorithm Inside-Outside Tests Scan -Line Fill of Curved Boundary Areas - Boundary-Fill Algorithm - Flood-Fill Algorithm - Overview of various attributes

Computer Graphics

Computer graphics refers to a technology that generates images on a computer screen. It's used in digital photography, film and television, video games, and on electronic devices and is responsible for displaying images effectively to users.

Computer Graphics Applications

- **Digital Element**

This can easily create any logo, cartoon, painting, frame, featured image, with help of computer graphics. Can create offline or online through computer software.

- **Medical Application**

The 2D colorful image of organs or the human body produced. Then this image is transferred to the CG tool. Surgeons use this for rehearse. X-ray, blood tests, any types of medical reports are also shown through graphics, Which makes it convenient for the doctor to give medicine to a patient.

- **Cartography**

This deals with the generation of informative maps and charts. Computer graphics are widely used to craft and design maps for different purposes. Computer graphics help to design the map focused on different strategies. As cartography combines science, aesthetics, and technique, it crafts maps so that users can easily communicate with maps and gather necessary information. Cartography was used in ancient times but in ancient times it is all designed by humans with hands-on stone or paper which was not accurate and informative. Nowadays, cartography uses computer graphics and printing material to design more precise and user-understandable maps showing information based on different aspects.

Computer Graphics Applications

- **Office Automation and Electronic Publishing**

Computers and Graphics systems are the main players who have changed the scenario of desktop publishing. Modern books, magazines, newspapers, and designers are successful in engaging more and more audiences. This application is mostly used by interior designers, engineers, architects, structural designers, etc. It makes them more powerful as they can design numerous components and systems. Going one step ahead from book and magazine publishing, computer graphics are used to design structures of automobile bodies, structures of buildings, airplanes, ships, optical instruments, and some major computer network systems.

- **Training**

Candidates may be trained using specialized training methods such as simulators to better understand them in a short period of time. Furthermore, Building training modules with computer graphics is straightforward and quite useful.

- **Machine Drawing**

Computer graphics are frequently uses for designing, modifying, and creating various parts of machines, as well as the entire machine. The main reason for this is that the precision and clarity we get from such drawings is ideal and highly desired for the safe manufacturing of machines using these drawings.

Computer Graphics Applications

IMAGE PROCESSING

- Image processing, applies techniques to modify or interpret existing picture such as photographs and TV scans.
- Two principal applications of image processing are (1) improving picture quality and (2) machine perception of visual information, as used in robotics.
- To apply image processing methods, digitize a photograph or other picture into an image file.
- Then digital methods can be applied to rearrange picture parts, to enhance color separations, or to improve the quality of shading.
- These techniques are used in commercial art applications that involve the retouching and rearranging of sections of photographs and other artwork.
- Similar methods are used to analyze satellite photos of the earth and photos of galaxies.
- Medical applications also make extensive use of image processing techniques for picture enhancements, in tomography and in simulations of operations.
- Tomography is a technique of X-ray photography that allows cross-sectional views of physiological systems to be displayed.
- Both computed X-rav tomography (CT) and position emission tomography (PET) use propchon methods to reconstruct cross sections from digital data.

Computer Graphics Applications

GRAPHICAL USER INTERFACES

- It is common now for software packages to provide a graphical interface.
- A major component of a graphical interface is a window manager that allows a user to display multiple-window areas.
- Each window can contain a different process that can contain graphical or nongraphical displays.
- To make a particular window active, we simply click in that window using an interactive pointing device.
- Interfaces also display menus and icons for fast selection of processing options
- or parameter values.
- An icon is a graphical symbol that is designed to look like the processing option it represents.
- The advantages of icons are that they take up less screen space than corresponding textual descriptions and they can be understood more quickly if well designed.
- Menus contain lists of textual descriptions and icons.
- The menus allow selection of processing options, color values, and graphics parameters.
- The icons represent options for painting, drawing, zooming, typing text strings, and other operations connected with picture construction.

Computer Graphics Applications

VISUALIZATION

- Scientists, engineers, medical personnel, business analysts, need to analyze large amounts of information or to study the behavior of certain processes.
- Numerical simulations carried out on supercomputers frequently produce data files containing thousands and even millions of data values.
- Similarly, satellite cameras and other sources are amassing large data files faster than they can be interpreted.
- Scanning these large sets of numbers to determine trends and relationships is a tedious and ineffective process.
- But if the data are converted to a visual form, the trends and patterns are often immediately apparent.
- Producing graphical representations for scientific, engineering, and medical data sets and processes is generally referred to as scientific visualization.
- And the term business visualization is used in connection with data sets related to commerce, industry,and other non scientific areas.
- There are many kinds of data sets, and effective visualization schemes depend on the characteristics of the data.
- A collection of data can contain scalar values, vectors, higher-order tensors, or any combination of these data types. And data sets can be two-dimensional or three dimensional.

Computer Graphics Applications

VISUALIZATION

- Color coding is just one way to visualize a data set.
- Additional techniques include contour plots, graphs and charts, surface renderings, and visualizations of volume interiors.
- In addition, image processing techniques are combined with computer graphics to produce many of the data visualizations.
- Mathematicians, physical scientists, and others use visual techniques to analyze mathematical functions and processes or simply to produce interesting graphical representations.
- A color plot of mathematical curve functions, and a surface plot of a function.
- Fractal procedures using quaternions generated the object, and a topological structure.
- Scientists are also developing methods for visualizing general classes of data.

Computer Graphics Applications

EDUCATION AND TRAINING

- Computer-generated models of physical, financial, and economic systems are often used as educational aids.
- Models of physical, physiological systems, population trends, or equipment, such as the color coded diagram can help trainees to understand the operation of the system.
- For some training applications, special systems(training of ship captains, aircraft pilots, heavy-equipment operators, and air traffic control personnel) are designed.
- Some simulators have no video screens. But most simulators provide graphics screens for visual operation.
- Another type of viewing system is viewing screen with multiple panels is mounted in front of the simulator and color projectors display the flight scene on the screen panels.
- Similar viewing systems are used in simulators for training aircraft control-tower personnel.
- The keyboard is used to input parameters affecting the airplane performance or the environment, and the pen plotter is used to chart the path of the aircraft during a training session.

Computer Graphics Applications

EDUCATION AND TRAINING

- An output from an automobile-driving simulator is used to investigate the behavior of drivers in critical situations.
- The drivers' reactions are then used as a basis for optimizing vehicle design to maximize traffic safety.

ENTERTAINMENT

- Computer graphics methods are commonly used in making motion pictures, music videos, and television shows.
- Sometimes the graphics scenes are displayed by themselves, and sometimes graphics objects are combined with the actors and live scenes.
- Many TV series regularly employ computer graphics methods.
- Music videos use graphics, Graphics objects can be combined with the live action or graphics and image processing techniques can be used to produce a transformation of one person or object into another (morphing).

Computer Graphics Applications

COMPUTER ART

- Computer graphics methods are widely used in both fine art and commercial art applications.
- Artists use a variety of computer methods, including special-purpose hardware, artist's paintbrush (such as Lumens), other paint packages, specially developed software, symbolic mathematics packages (such as Mathematics), CAD packages, desktop publishing software, and animation packages that provide facilities for designing object shapes and specifying object motions.
- A paintbrush system, with pressure-sensitive stylus, was used to produce the electronic painting that simulates the brush strokes.
- The stylus translates changing hand pressure into variable line widths, brush sizes, and color gradations.
- Fine artists use a variety of other computer technologies to produce images.
- This artist uses a combination of mathematical functions, fractal procedures, Mathematics software, ink-jet printers, and other systems to create a variety of three-dimensional and two-dimensional shapes and stereoscopic image pairs.
- For many applications of commercial art (and in motion pictures and other applications), photorealistic techniques are used to render images of a product.

Computer Graphics Applications

COMPUTER ART

- Animations are also used frequently in advertising, and television commercials are produced frame by frame, where each frame of the motion is rendered and saved as an image file.
- In each successive frame, the motion is simulated by moving object positions slightly from their positions in the previous frame.
- When all frames in the animation sequence have been rendered, the frames are transferred to film or stored in a video buffer for playback.
- Film animations require 24 frames for each second in the animation sequence.
- If the animation is to be played back on a video monitor, 30 frames per second are required.
- A common graphics method employed in many commercials is morphing, where one object is transformed (metamorphosed) into another.
- This method has been used in TV commercials to an oil can into an automobile engine, an automobile into a tiger, a puddle of water into a tire, and one person's face into another face.

Computer Graphics Applications

PRESENTATION GRAPHICS

- Presentation graphics is used to produce illustrations for reports or to generate 35-mm slides or transparencies for use with projectors.
- Presentation graphics is commonly used to summarize financial, statistical, mathematical, scientific, and economic data for research reports, manage rial reports, consumer information bulletins, and other types of reports.
- Workstation devices and service bureaus exist for converting screen displays into 35-mm slides or overhead transparencies for use in presentations.
- Typical examples of presentation graphics are bar charts, line graphs, surface graphs, pie charts, and other displays showing relationships between multiple parameter.
- The two-dimensional graphics combined with geographical information.
- Similar graphs and charts can be displayed in three dimensions to provide additional information.
- Three-dimensional graphs are sometime used simply for effect; they can provide a more dramatic or more attractive presentation of data relationships.
- The charts include a three-dimensional bar graph and an exploded pie chart.

Computer Graphics Applications

COMPUTER AIDED DESIGN (CAD)

- Engineering and architectural systems are designed (Buildings, automobiles, aircraft, watercraft, spacecraft, textiles).
- For some design applications; object are displayed in a wireframe outline form that shows the overall sham and internal features of objects.
- Wireframe displays also allow designers to quickly see the effects of interactive adjustments to design shapes.
- Software packages for CAD applications typically provide the designer with a multi-window environment.
- The windows show enlarged sections or different views of objects.
- Shapes used in a design represent the different network or circuit components.
- Standard shapes for electrical, electronic, and logic circuits are often supplied by the design package.
- For other applications, a designer can create personalized symbols that are to be used to construct the network or circuit.
- The system is then designed by successively placing components into the layout, with the graphics package automatically providing the connections between components.
- This allows the designer to quickly try out alternate circuit schematics for minimizing the number of components or the space required for the system.

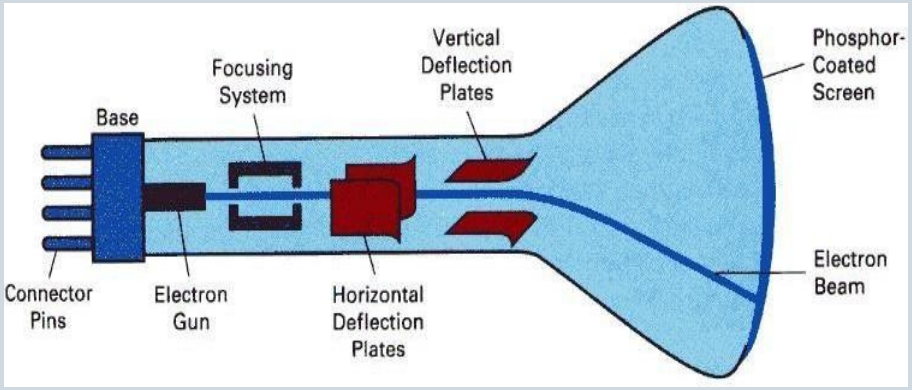
Graphics System Overview:

Video Display Devices

- Refresh Cathode-Ray Tubes
- Raster-Scan Displays
- Random-Scan Displays
- Color CRT Monitors
- Direct View Storage Tubes
- Flat-Panel Displays

Graphics System

Refresh Cathode-Ray Tubes (CRT) - still the most common video display device presently



Electrostatic deflection of the electron beam in a CRT

An electron gun emits a beam of electrons, which passes through focusing and deflection systems and hits on the phosphor-coated screen. The number of points displayed on a CRT is referred to as r**esolutions** (eg. 1024x768). Different phosphors emit small light spots of different colors, which can combine to form a range of colors. A common methodology for color CRT display is the **Shadow-mask** meth

The light emitted by phosphor fades very rapidly, so it needs to redraw the picture repeatedly.
There are 2 kinds of redrawing mechanisms:
Raster-Scan and Random-Scan

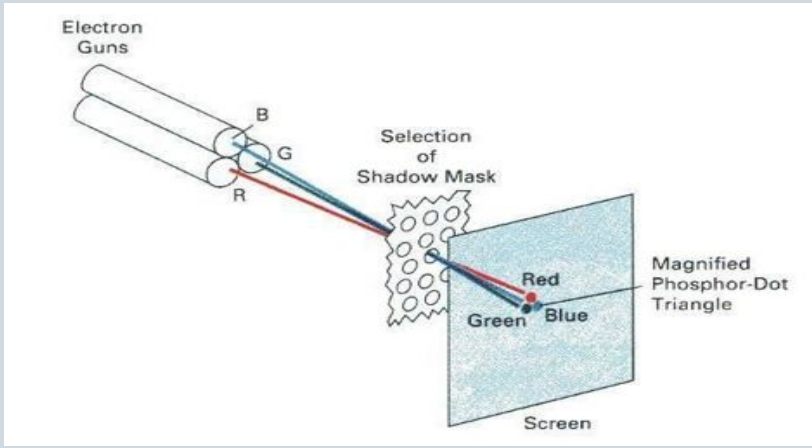


Illustration of a shadow-mask CRT

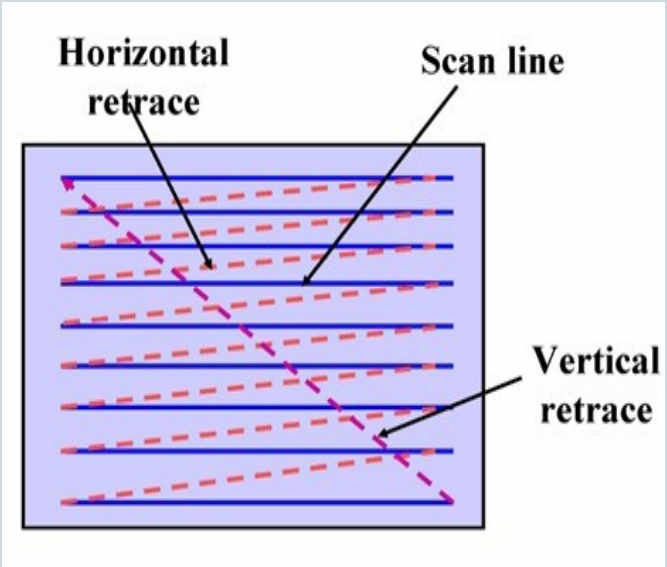
Graphics System Overview

Raster-Scan Displays

- The electron beam is swept across the screen one row at a time from top to bottom. Each row is referred to as a scan line.
- Picture definition is stored in the frame buffer. This memory area holds the set of intensity values for the screen points. These stored values are then retrieved from the refresh buffer and used to control the intensity of the electron beam as it moves from spot to spot across the screen.

Refreshing on raster-scan display is carried out at the rate of 60-80 frames per seconds, this can be done by using following retrace techniques

- Horizontal retrace
- Vertical retrace



Graphics System Overview

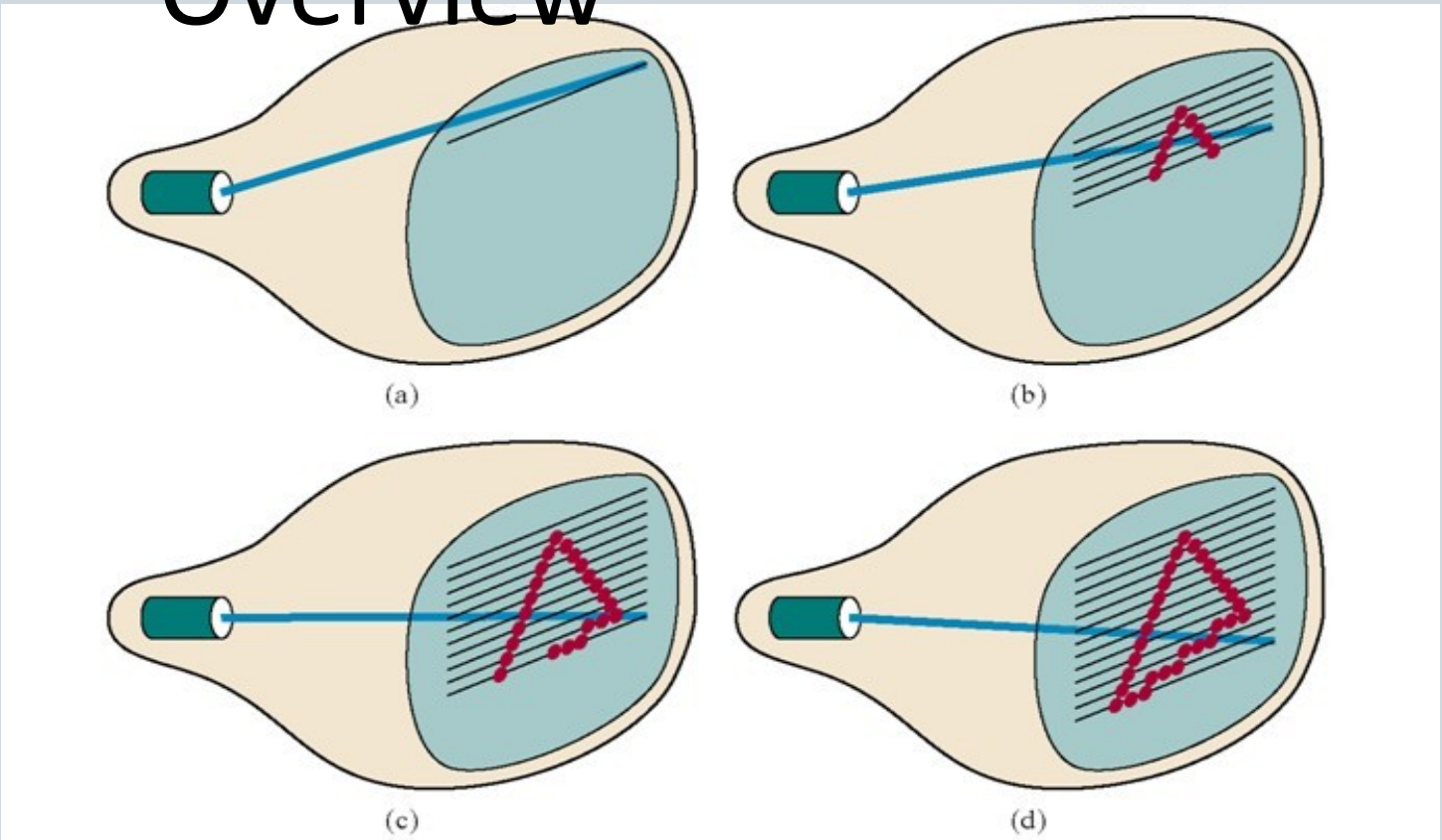
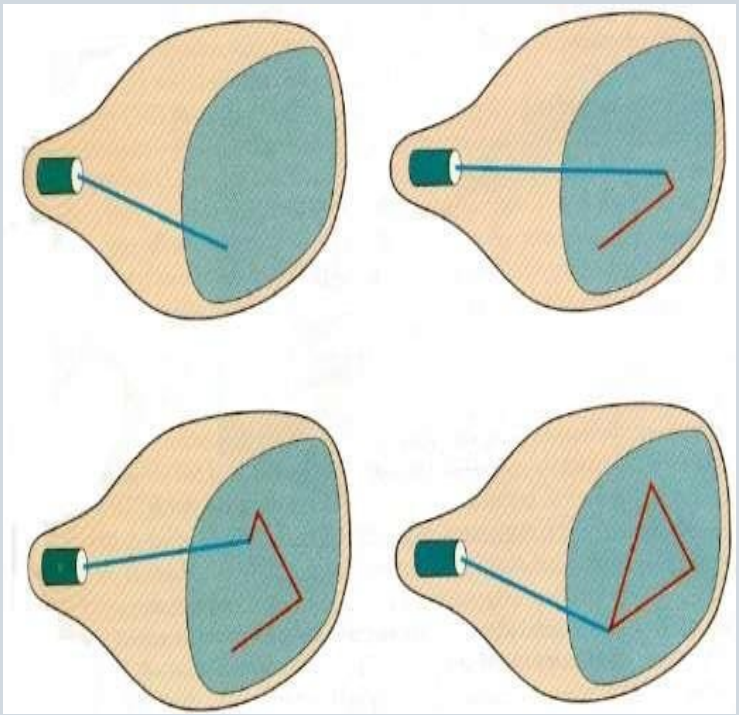


Figure 2-7
A raster-scan system displays an object as a set of discrete points across each scan line.

Graphics System

Raster-Scan Displays Random-Scan Displays



In this technique, the electron beam is directed only to the part of the screen where the picture is to be drawn rather than scanning from left to right and top to bottom as in raster scan. It is also called **vector display, stroke-writing display,** or **calligraphic display**.

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

Graphics System Overview

Color CRT Monitor: It is similar to a CRT monitor.

The basic idea behind the color CRT monitor is to combine three basic colors- Red, Green, and Blue. By using these three colors, we can produce millions of different colors.

The two basic color display producing techniques are:

Beam penetration method

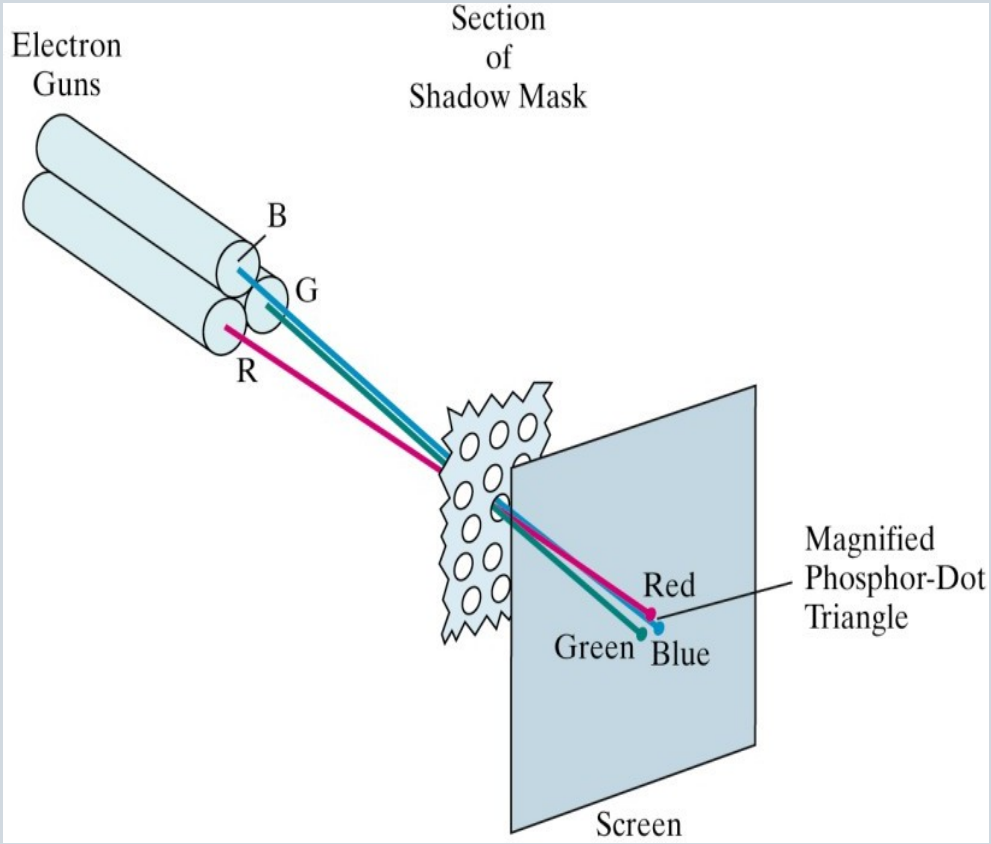
Shadow mask method

Graphics System Overview

Beam penetration method

Random scan monitors use the beam penetration method for displaying color picture. In this, the inside of CRT screen is coated two layers of phosphor namely red and green.

A beam of slow electrons excites only the outer red layer, while a beam of fast electrons penetrates red layer and excites the inner green layer. At intermediate beam speeds, combination of red and green light are emitted to show two additional colors- orange and yellow.



Advantages

Less expensive

Disadvantages

Quality of images are not good as comparatable with other methods

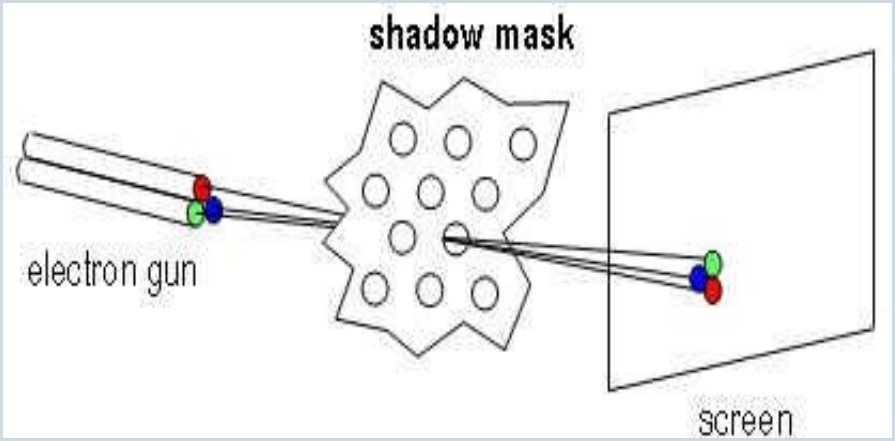
Graphics System

Overview

Shadow Mask Method

Raster scan system are use shadow mask methods to produced a much more range of colors than beam penetration method.

In this, CRT has three phosphor color dots. One phosphor dot emits a red light, second emits a green light and third emits a blue light.



Advantage:
Realistic image
Million different colors to be generated
Shadow scenes are possible

Disadvantage:
Relatively expensive compared with the monochrome CRT.
Relatively poor resolution
Convergence Problem

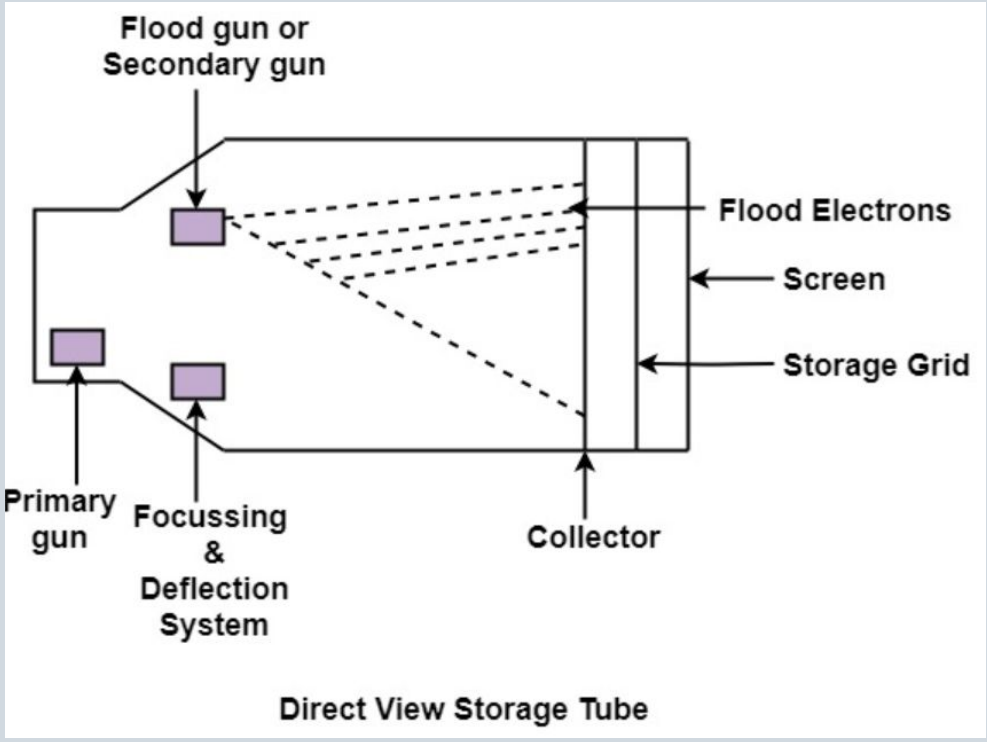
Graphics System Overview

Direct View Storage Tubes

DVST terminals also use the random scan approach to generate the image on the CRT screen. The term "storage tube" refers to the ability of the screen to retain the image which has been projected against it, thus avoiding the need to rewrite the image constantly.

Function of guns: Two guns are used in DVST

1. **Primary guns:** It is used to store the picture pattern.
2. **Flood gun or Secondary gun:** It is used to maintain picture display.



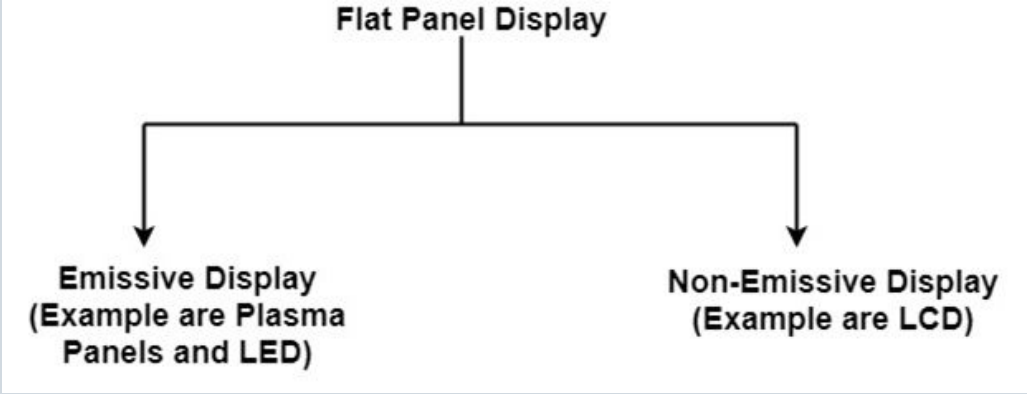
Direct View Storage Tube

Graphics System Overview

Flat Panel Display

The Flat-Panel display refers to a class of video devices that have reduced volume, weight and power requirement compare to CRT.

Example: Small T.V. monitor, calculator, pocket video games, laptop computers, etc



Emissive Display: The emissive displays are devices that convert electrical energy into light. Examples are Plasma Panel, thin film electroluminescent display and LED (Light Emitting Diodes).

2. Non-Emissive Display: The Non-Emissive displays use optical effects to convert sunlight or light from some other source into graphics patterns. Examples are LCD (Liquid Crystal Device).

Graphics System Overview:

Feature	Description of Raster-Scan Displays
Technology Base	Based on television technology, employing a CRT.
Electron Beam Movement	Swept across the screen from top to bottom, one row at a time.
Intensity Control	Beam intensity turned on and off to create a pattern of illuminated spots.
Memory Storage	Stored in a **refresh buffer/frame buffer** holding intensity values for all points. Stored intensity values are retrieved from the refresh buffer and "painted" on the screen one row (scan line) at a time.
Pixel Definition	Each screen point is a pixel or pel(Pixel element), allowing for the display of scenes with subtle shading and color patterns.
Color and Intensity Variation	Varies from 1 bit in black-and-white systems to 24 bits per pixel in high-quality systems. The size depends on resolution. • For one bit per pixel, frame buffer is -**BITMAP**. • For systems with multiple bits per pixel, frame buffer - **PIXMAP**.
Refresh Rate	**Refresh rates** are units of cycles per second, or Hertz (Hz), where a cycle corresponds to one frame. 60 to 80 fps, with higher rates possible.
Horizontal Retrace	After each scan line, the beam returns to the left side of the screen.
Vertical Retrace	After each frame(displayed in 1/80th - 1/60th of a second), the beam returns to the top left corner of the screen to begin the next frame..
Interlaced Refresh	Used in systems to display screen in 2 passes, reducing flicker at low refresh rates. • First pass- beam sweeps across every other scan line from top to bottom. • After the vertical retrace, the beam sweeps out the remaining scan lines . • Interlacing of the scan lines allows to see the entire screen displayed in one-half the time it would have taken to sweep across all the lines at once from top to bottom. Interlacing is primarily used with slower refreshing rates.

Graphics System Overview:

Feature	Description of Random-Scan Displays
Beam Direction	Directed only to the parts of the screen where a picture is to be drawn.
Display Type	**vector / stroke-writing displays**, draw a picture one line at a time.
Refresh Method	The system cycles through the set of line drawing commands in the refresh display file/display list/display program.
Refresh Rate	Depends on the number of lines; typically, 30 to 60 times each second for high-quality systems.
Memory Storage	Picture definition is stored as line drawing commands in an area of memory referred to as the refresh display file.
Image Quality	Capable of producing smooth line drawings with higher resolution
Common Devices	Pen plotters and other devices that require line drawing capabilities.
Limitations	Cannot display realistic shaded scenes; not suitable for applications requiring intense color variations. Since picture definition is stored as a set of line drawing instructions and not as a set of intensity values for all screen points
Display List Processing	After processing all commands, the system cycles back to the first command in the display list.
Phosphor Burnout Prevention	Refresh cycles may be delayed to prevent burnout of the phosphor from too high a refresh rate.
Line Drawing Quality	Produces smooth line drawings as the CRT beam directly follows the line path.

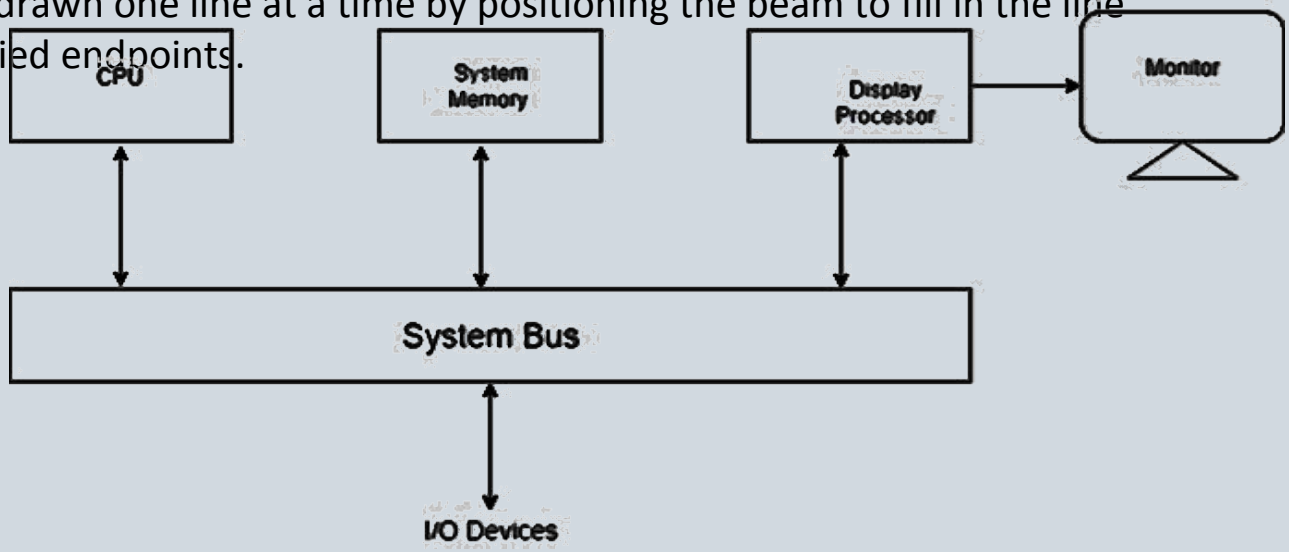
Graphics System

Characteristic	Random-Scan Displays	Raster-Scan Displays
Beam Direction	Directed only to parts where a picture is drawn.	Swept across the screen, row by row from top to bottom.
Drawing Method	Draws one line at a time (vector or stroke- writing displays).	Creates a pattern of illuminated spots (pixel-based).
Refresh Method	Cycles through a set of line drawing commands.	Retrieves and paints stored intensity values for each pixel in sequence.
Refresh Rate	30 to 60 times per second, adjusted based on the number of lines.	60 to 80 frames per second, higher with interlaced refresh.
Memory Storage	Line drawing commands in the refresh display file or buffer.	Intensity values for each pixel in the frame buffer.
Image Quality	High resolution with smooth line drawings.	Can display complex color patterns, may have aliasing issues.
Applications	Line drawing applications, not suitable for shaded images.	Suitable for realistic display of scenes with shading and color.
Phosphor Burnout Prevention	Refresh rates may be delayed to prevent burnout.	Constant refresh rate; interlacing can prevent burnout.
Color and Shading	No shaded scenes, primarily line drawings.	Capable of displaying shaded scenes and color variations.
Resolution	Generally higher resolution than raster systems.	Resolution depends on the number of pixels and refresh rate.
Common Devices	Pen plotters and devices that require line drawing.	Home television sets, monitors, and printers.

Graphics System Overview:

RANDOM-SCAN SYSTEMS

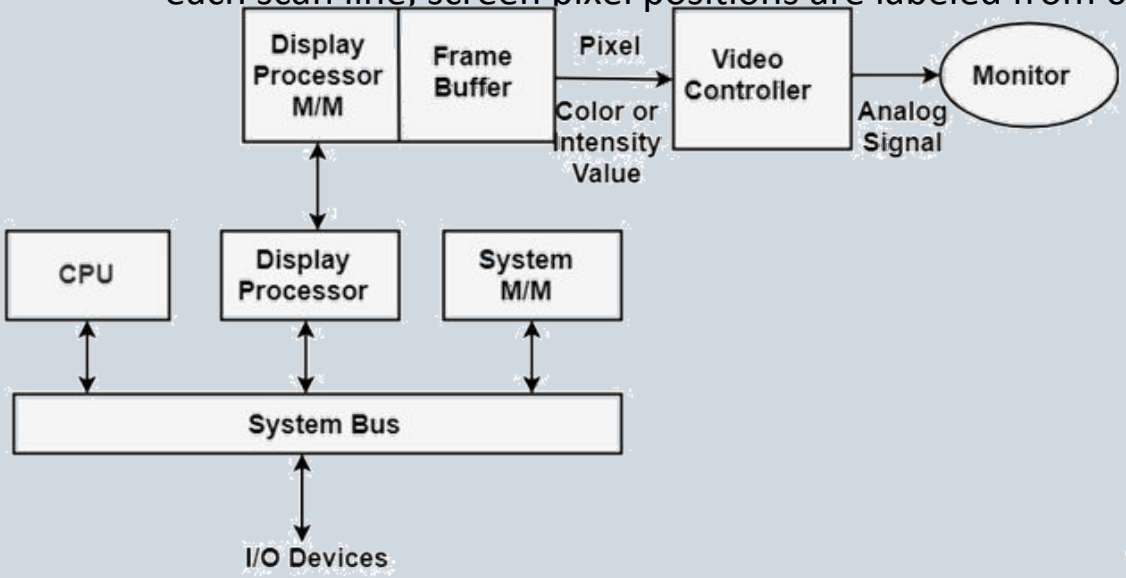
- An application program is input and stored in the system memory along with a graphics package.
- Graphics commands in the application program are translated by the graphics package into a display file stored in the system memory.
- This display file is then accessed by the display processor to refresh the screen.
- The display processor(display processing unit/graphics controller) cycles through each command in the display file program once during every refresh cycle.
- Graphics patterns are drawn by directing electron beam along component lines of picture.
- Lines are defined by the values for their coordinate endpoints, and these input coordinate values are con- verted 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.

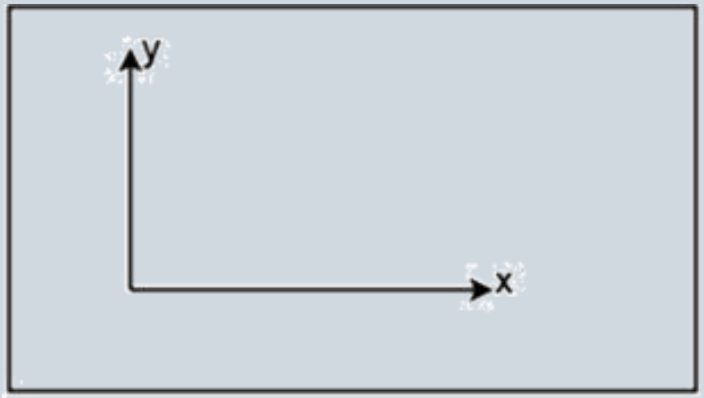


Graphics System Overview:

RASTER-SCAN SYSTEMS

- Interactive raster graphics systems employ central processing unit(CPU), a video controller or display controller, is used to control the operation of the display device.
- Video controller accesses frame buffer which is in system memory to refresh the screen.
- Frame-buffer locations, and the corresponding screen positions, are referenced in Cartesian coordinates.
- For many graphics monitors, the coordinate origin is defined at lower left screen comer.
- The screen surface is represented as the first quadrant of a 2D system, with +x values increasing to the right and +y values increasing from bottom to top.
- Scan lines are then labeled from ymax, at the top of the screen to 0 at the bottom. Along each scan line, screen pixel positions are labeled from 0 to xmax.





Graphics System Overview:

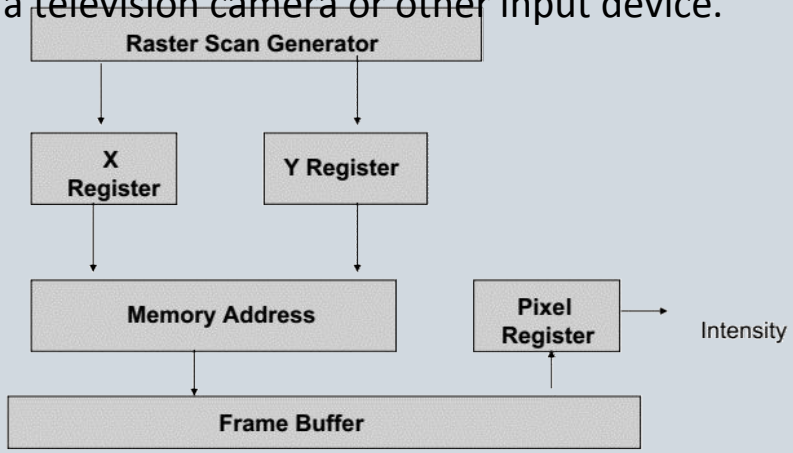
RASTER-SCAN SYSTEMS

- Two registers (x =0 and the y =Ymax) are used to store coordinates of the screen pixels.
- The value stored in the frame buffer for this pixel position is then retrieved and used to set the intensity of the CRT beam.
- X++, and the process is repeated for each pixel along scan line.
- After the last pixel on the top scan line has been processed, the x is reset to 0 and the Y--.
- Pixels along this scan line are then processed, and the procedure is repeated for each successive scan line.
- After cycling through all pixels along bottom scan line (y=O), 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 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.
- A few other operations can be performed by the video controller, besides the basic refreshing operations.
- For various applications, the video controller can retrieve pixel intensities from different memory areas on different refresh cycles.

Graphics System Overview:

RASTER-SCAN SYSTEMS

- In high-quality systems, for example, two frame buffers are provided so that one buffer can be used for refreshing while the other is being filled with intensity values.
- Then the two buffers can switch roles.
- This provides a fast mechanism-for generating real-time animations, since different views of moving objects can be successively loaded into the refresh buffers.
- Some transformations can be accomplished by the video controller.
- Areas of the screen can be enlarged, reduced, or moved from one location to another during the refresh cycles.
- The video controller often contains a lookup table, so that pixel values in the frame buffer are used to access the lookup table instead of controlling the CRT beam intensity directly.
- This provides a fast method for changing screen intensity values.
- Some systems are designed to allow the video controller to mix the frame-buffer image with an input image from a television camera or other input device.



Graphics System Overview:

RASTER-SCAN DISPLAY PROCESSOR

- The display processor free the CPU from the graphics chores.
- Display processor digitizes 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.
- Graphics commands specifying straight lines and other geometric objects are scan converted into a set of discrete intensity points.
- Scan converting a straight-line segment, for example, means that we have to locate the pixel positions closest to the line path and store the intensity for each position in the frame buffer.
- Similar methods are used for scan converting curved lines and polygon outlines.
- Characters can be defined with rectangular grids or they can be defined with curved outlines.
- The array size for character grids can vary from about 5 * 7 to 9 * 12 or more for higher-quality displays.
- A character grid is displayed by superimposing the rectangular grid pattern into the frame buffer at a specified coordinate position.
- With characters that are defined as curve outlines, character shapes are scan converted into the frame buffer.
- Display processors perform functions like generating various line styles (dashed, dotted, or solid), displaying color areas, and performing certain transformations and manipulations on displayed objects.

Graphics System Overview:

RASTER-SCAN DISPLAY PROCESSOR

- Display processors designed to interface with input devices(mouse).
- In an effort to reduce memory requirements in raster systems, methods have been devised for organizing the frame buffer as a linked list and encoding curve outline. the intensity information.
- One way to do this is to store each scan line as a set of integer pairs.
- One number of each pair indicates an intensity value, and the 2nd specifies the number of adjacent pixels on the scan line that are to have that intensity. This technique is run-length encoding, can result in saving storage space if a picture is to be constructed mostly with long runs of a single color each.
- A similar approach can be taken when pixel intensities change linearly.
- Another approach is to encode the raster as a set of rectangular areas (cell encoding). The disadvantages of encoding runs are that intensity changes are difficult to make and storage requirements actually increase as the length of the runs decreases.
- In addition, it is difficult for the display controller to process the raster when many short runs are involved.

Graphics System

Graphic Monitors and Workstations

Overview:

Graphics Monitor:

i).The name itself giving some idea that, Monitor which is capable of displaying "Graphics". Generally most of the Monitors support Text modes.

ii).So Monitors which are capable of showing and supporting Graphics mode along with the Text modes.

iii).Graphic monitors who can display pictures on its screen, of course it acts like an output device. The monitors which support the following Graphic applications are said to be Graphic Monitors.

The Graphics Applications include,

A).Animation Software's

B).CAD Software's

C).Drawing programs

D).Paint Application programs

E).Presentation Graphics Software's (Excel likewise, creating pie and bar charts)

F).Desktop publishing (MS Office, Share points, Document managements)

So by now, you might have got an Idea of what is Graphic Monitor Actually is??

Graphics System Overview:

Graphic Monitors and Workstations

Graphics Workstations:

i).Workstation is also a computer which varies generally with other General Computers

ii).Because these Workstations need good operating power of computer. They must be able to support high power i.e. it must sustain good graphic capabilities.

iii).These kind of computers comes with the following specifications. Unlike normal general computers they consists of,

A).Minimum of 64 Megabytes of RAM

B).Very good Resolution Graphics screen

C).High and Large Screen

D).GUI Graphical User Interface (which helps programs easier to use of the Computer's Graphics)

E).Very good Mass storage Device like Disk Drive

F).Built-in Network Support and many factors

We may also notice that, some of the workstations do not have any disk drives in it. So these kind of disk drives are called as Diskless workstation.

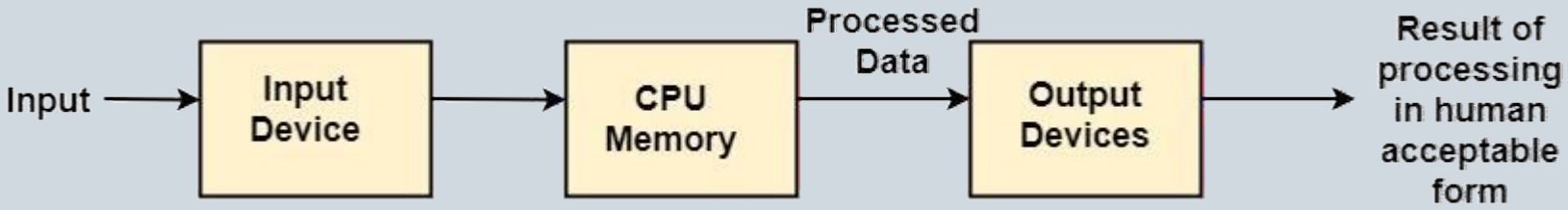
Workstations took place of and they lie between Personal computer (General Computers) and minicomputers as far as Computing power is concerned.

These can be used as both types such as stand system (which only consists of one) and Local area network. So basically in this LAN kind, workstations are typically connected together one and other.

Graphics System Overview:

Input devices

- Keyboard,
- Mouse,
- Trackball/Spaceball ,
- Joystick,
- Digitizers,
- Data gloves,
- Touch panels,
- Image scanners,
- Light pens,
- Voice systems



Graphics System Overview:

Input devices

Keyboards

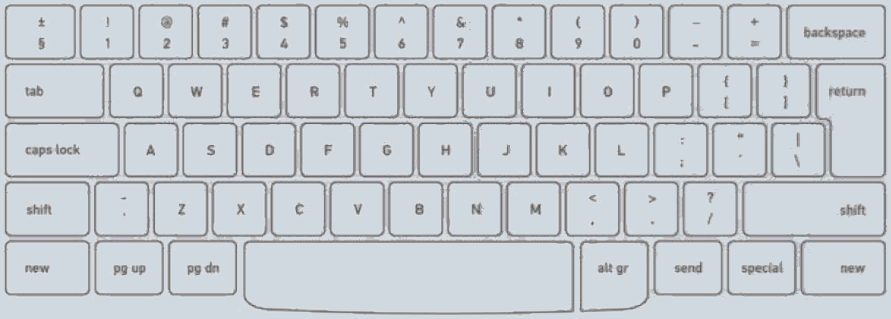
- An alphanumeric keyboard is used for entering text strings.
- The keyboard is for inputting such nongraphic data as picture labels associated with a graphics display.
- Keyboards can also be provided with features to facilitate entry of screen coordinates, menu selections, or graphics functions.
- Cursor-control keys and function keys are common features on general purpose keyboards.
- Function keys allow users to enter frequently used operations in a single keystroke, and cursor-control keys can be used to select displayed objects or coordinate positions by positioning the screen cursor.
- Other cursor-positioning devices(trackball or joystick) are included on some keyboards.
- Additionally, a numeric keypad is,often included on the keyboard for fast entry of numeric data.
- For specialized applications, input to a graphics application may come from a set of buttons, dials, or switches that select data values or customized graphics operations.

Graphics System Overview:

Input devices

Keyboards

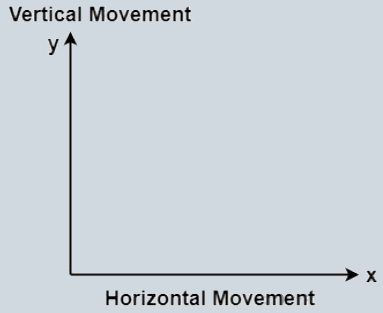
- Buttons and switches are often used to input predefined functions, and dials are common devices for entering scalar values.
- Real numbers within some defined range are selected for input with dial rotations.
- Potentiometers are used to measure dial rotations, which are then converted to deflection voltages for cursor movement.



Graphics System Overview:

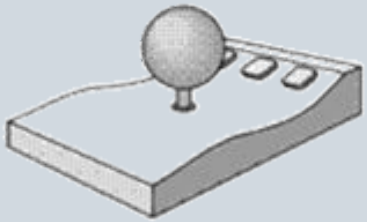
Input devices

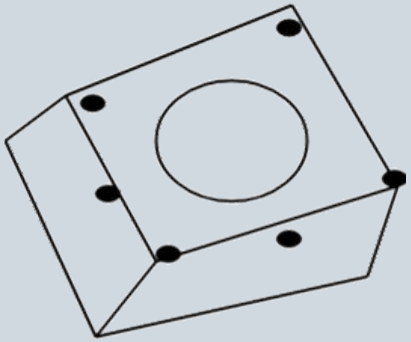
Mouse



- A mouse is small hand-held box used to position the screen cursor.
- Wheels or rollers on the bottom of the mouse can be used to record the amount and direction of movement.
- Another method for detecting mouse motion is with an optical sensor.
- The mouse is moved over a mouse pad with grid of horizontal and vertical lines.
- The optical sensor detects movement across the lines in the grid.
- Since a mouse can be picked up and put down at another position without change in cursor movement, it is used for making relative changes in the position of the screen cursor.
- Three buttons are included on the top of the mouse for signaling the execution of some operation, such as recording cursor position or invoking a function.
- Z mouse includes three buttons, a thumbwheel on the side, a trackball on the top, and a standard mouse ball underneath.
- This design provides six degrees of freedom to select Input Devices spatial positions, rotations, and other parameters.
- With Z mouse, an object can be picked up , rotated, and moved in any direction, or navigated viewing position and orientation through a 3D scene.
- Applications of the Z mouse include virtual reality, CAD, and animation.

Graphics System Overview:



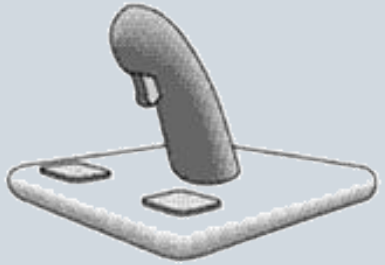


Input devices

Trackball and Spaceball

- Trackball is a ball that can be rotated with the fingers or palm of the hand, to produce screen-cursor movement.
- Potentiometers, attached to the ball, measure the amount and direction of rotation.
- Trackballs are often mounted on keyboards or other devices such as Z mouse.
- While a trackball is a two-dimensional positioning device, a Spaceball provides six degrees of freedom.
- Unlike the trackball, a Spaceball does not actually move.
- Strain gauges measure the amount of pressure applied to the Spaceball to provide input for spatial positioning and orientation as the ball is pushed or pulled in various directions.
- Spaceballs are used for three-dimensional positioning and selection operations in virtual-reality systems, modeling, animation, CAD, and other applications.

Graphics System Overview:



Input devices

Joysticks

- A joystick consists of a small, vertical lever (called the stick) mounted on a base that is used to steer the screen cursor around.
- Joysticks select screen positions with actual stick movement; others respond to pressure on the stick.
- Joysticks are mounted on a keyboard; others function as stand-alone units.
- The distance that the stick is moved in any direction from its center position corresponds to screen-cursor movement in that direction.
- Potentiometers mounted at the base of the joystick measure the amount of movement, and springs return stick to the center position when it is released.
- One or more buttons can be programmed to act as input switches to signal certain actions once a screen position has been selected.
- The movable joystick, is used to activate switches that cause the screen cursor to move at a constant rate in the direction selected.
- 8 switches, arranged in a circle, are sometimes provided, so that the stick can select any one of eight directions for cursor movement.
- Pressure sensitive joysticks (isometric joysticks), have a non-movable stick.
- Pressure on the stick is measured with strain gauges and converted to movement of the cursor in the direction specified.

Graphics System Overview:

Input devices

Data Glove

- Data glove is used to grasp a "virtual" object.
- The glove is constructed with a series of sensors that detect hand and finger motions.
- Electromagnetic coupling between transmitting antennas and receiving antennas is used to provide information about the position and orientation of the hand.
- The transmitting and receiving antennas can each be structured as a set of three mutually perpendicular coils, forming a three-dimensional Cartesian coordinate system.
- Input from the glove can be used to position or manipulate objects in a virtual scene.
- A two-dimensional projection of the scene can be viewed on a video monitor, or a three-dimensional projection can be viewed with a headset.

Graphics System Overview:

Input devices

Digitizers

- For drawing, painting, or interactively selecting coordinate positions on an object is a digitizer.
- These devices can be used to input coordinate values in a 2D/3D space.
- Digitizer is used to scan over a drawing or object and to input a set of discrete coordinate positions, which can be joined with straight-line segments to approximate the curve or surface shapes.
- Graphics tablet (data tablet), is used to input 2D coordinates by activating a hand cursor/stylus at selected positions on a flat surface.
- A hand cursor contains cross hairs for sighting positions, while a stylus is a pencil-shaped device that is pointed at positions on the tablet.
- The artist's digitizing system uses electromagnetic resonance to detect the three-dimensional position of the stylus.
- This allows an artist to produce different brush strokes with different pressures on the tablet surface.
- Tablet size varies 12 * 12 in for desktop to 44 * 60 in or larger for floor models.
- Graphics tablets provide a highly accurate method for selecting coordinate positions, with an accuracy that varies from about 0.2 mm on desktop models to about 0.05 mm or less on larger models.

Graphics System Overview:

Input devices

Digitizers

- Many graphics tablets are constructed with a rectangular grid of wires embedded in the tablet surface.
- Electromagnetic pulses are generated in sequence along the wires, and an electric signal is induced in a wire coil in an activated stylus or hand cursor to record a tablet position.
- Depending on the technology, either signal strength, coded pulses, or phase shifts can be used to determine the position on the tablet.
- Acoustic (or sonic) tablets use sound waves to detect a stylus position.
- Either strip microphones or point microphones can be used to detect the sound emitted by an electrical spark from a stylus tip.
- The position of the stylus is calculated by timing the arrival of the generated sound at the different microphone 2-5 positions.
- An advantage of 2D acoustic tablets is that the micro- Input Devices phones can be placed on any surface to form the "tablet" work area.
- This can be convenient for various applications, such as digitizing drawings in a book.

Graphics System Overview:

Input devices

Digitizers

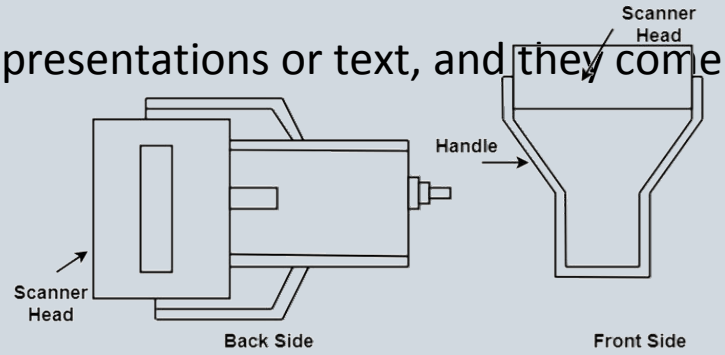
- 3D digitizers use sonic or electromagnetic transmissions to word positions.
- One electromagnetic transmission method is like that used in the data glove: A coupling between the transmitter and receiver is used to compute the location of a stylus as it moves over the surface of an obpct.
- As the points are selected on a nonmetallic object, a wireframe outline of the surface is displayed on the computer screen.
- Once the surface outline is constructed, it can be shaded with lighting effects to produce a realistic display of the object.
- Resolution of this system is h m 0.8 mm to 0.08 mm, depending on the model.

Graphics System Overview:

Input devices

Image Scanners

- Drawings, graphs, color and black-and-white photos, or text can be stored for computer processing with an image scanner by passing an optical scanning mechanism over the information to be stored.
- The gradations of gray scale or color are then recorded and stored in an array.
- Once we havethe internal representation of a picture, we can apply transformations to rotate, scale, or crop the picture to a particular screen area.
- We can also apply various image-processing methods to modify the array representation of the picture.
- For scanned text input, various editing operations can be performed on the stored documents.
- Some scanners can scan either graphical representations or text, and they come in a variety of sizes and capabilities.



Graphics System Overview:

Input devices

Touch Panels

- As the name implies, touch panels allow displayed objects or screen positions to be selected with the touch of a finger.
- A typical application of touch panels is for the selection of processing options that are represented with graphical icons.
- Some systems, such as the plasma panels are designed with touch screens.
- Other systems can be adapted for touch input by fitting a transparent device with a touch sensing mechanism over the video monitor screen.
- Touch input can be recorded using optical, electrical, or acoustical methods.
- Optical touch panels employ a line of infrared light-emitting diodes (LEDs) along one vertical edge and along one horizontal edge of the frame.
- The opposite vertical and horizontal edges contain light detectors.
- These detectors are used to record which beams are interrupted when the panel is touched.
- The two crossing beams that are interrupted identify the horizontal and vertical coordinates of the screen position selected.
- Positions tin be selected with an accuracy of about ¼ inch.
- With closely spaced LEDs, it is possible to break 2 horizontal or 2 vertical beams simultaneously.

Graphics System Overview:

Input devices

Touch Panels

- Average position between the two interrupted beams is recorded.
- The LEDs operate at infrared frequencies, so that the light is not visible to a user.
- The arrangement of LEDs in an optical touch panel that is designed to match the color and contours of the system to which it is to be fitted.
- An electrical touch panel is constructed with two transparent plates separated by a small distance.
- One of the plates is coated with a conducting material, and the other plate is coated with a resistive material.
- When the outer plate is touched, it is forced into contact with the inner plate.
- This contact creates a voltage drop across the resistive plate that is converted to the coordinate values of the selected screen position.
- In acoustical touch panels, high-frequency sound waves are generated in the horizontal and vertical directions across a glass plate.
- Touching screen causes part of each wave to be reflected from finger to emitters.
- The screen position at the point of contact is calculated from a measurement of the time interval between the transmission of each wave and its reflection to the emitter.

Graphics System Overview:

Input devices

Light Pens

- Light pen are pencil-shaped devices used to select screen positions by detecting the light coming from points on the CRT screen.
- They are sensitive to the short burst of light emitted from the phosphor coating at the instant the electron beam strikes a particular point.
- Other Light sources, such as the background light in the room, are usually not detected by a light pen.
- An activated light pen, pointed at a spot on the screen as the electron beam lights up that spot, generates an electrical pulse that causes the coordinate position of the electron beam to be recorded.
- As with cursor-positioning devices, recorded Light-pen coordinates can be used to position an object or to select a processing option.
- For one, when a light pen is pointed at the screen, part of the screen image is obscured by the hand and pen.
- Also, light pens require special implementations for some applications because they cannot detect positions within black areas.
- To be able b select positions in any screen area with a light pen, we must have some nonzero intensity assigned to each screen pixel.
- Light pens sometimes give false readings due to background lighting in a room.

Graphics System Overview:

Input devices

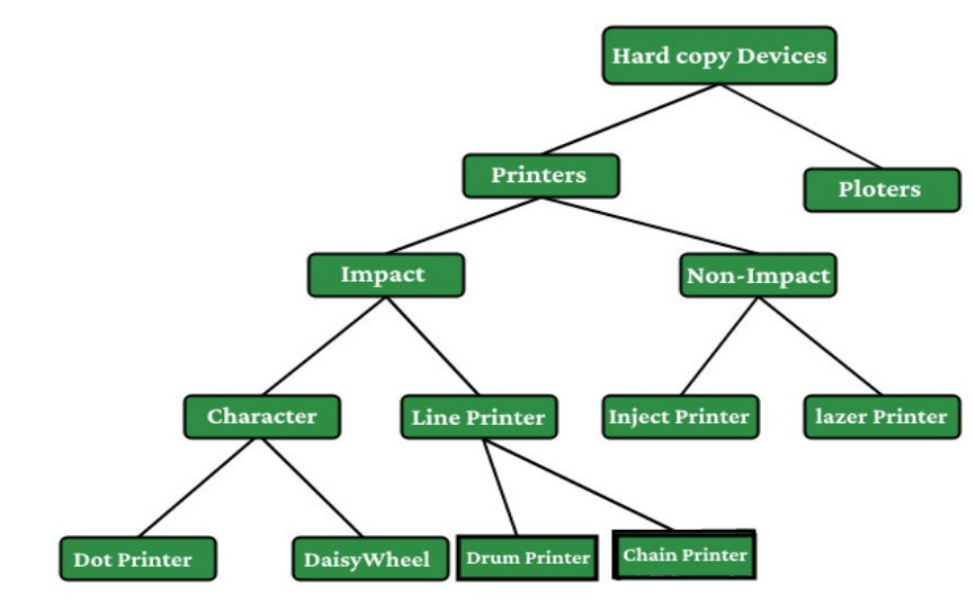
Voice Systems

- Speech recognizers are used in some graphics workstations as input devices to accept voice commands.
- Voice-system input can be used to initiate graphics operations or to enter data.
- These systems operate by matching an input against a predefined dictionary of words and phrase.
- A dictionary is set up for a particular operator by having, the operator speak the command words to be used into the system.
- Each word is spoken several times, and the system analyzes the word and establishes a frequency pattern for that word in the dictionary along with the corresponding function to be performed.
- When a voice command is given, the system searches the dictionary for a frequency-pattern match.
- Voice input is typically spoken into a microphone mounted on a headset.
- The microphone is designed to minimize input of other background sounds.
- If a different operator is to use the system, the dictionary must be reestablished with that operator's voice patterns.

Graphics System Overview:

Hard-Copy Devices

A hard copy is a printed copy of information from a computer. Sometimes it refers to a printer, so it is called a hard copy because it exists as a physical object. A hard copy is a tangible output that is usually printed. The principal examples are printouts, whether text on graphics, form printers, and also films including microfilms and microfiche is also considered as hardcopy output.



Graphics System

Hard-Copy Devices Overview:

Printers:

The printer is the most important output device, which is used to print data on paper. A printer is an important accessory for any computer system, especially for a graphics system. This is because most of the graphics created using computer graphics have their ultimate utilization in printed form. Printers are of two types impact and non-impact printers.

There are several major printer technologies available. These technologies can be broken down into two main categories with several types in each. Impact printers involve mechanical components for conducting printing. While in non-impact printers, no mechanical moving components are used.

Plotters:

A plotter is a special output device used to produce hard copies of large graphics and designs on paper, such as contributed maps, and engineering drawings. The plotter is either a peripheral component that you add to your computer system with its own internal processor. It is suitable for applications:

- Architectural plan of the building.
- CAD applications like the design of mechanical components of aircraft.
- Many engineering applications.

Graphics System Overview:

Hard-Copy Devices

Dot Matrix Printer

Daisy Wheel Printer

Inkjet Printer

Laser Printer

Graphics System Overview:

Graphic Software:

There are mainly two types of graphics software:

1. General programming package
2. Special-purpose application package

General programming package

A general programming package provides an extensive set of graphics function that can be used in high level programming language such as C or FORTRAN.

It includes basic drawing element shape like line, curves, polygon, color of element transformation etc.

Example: - GL (Graphics Library).

Special-purpose application package

Special-purpose application package are customize for particular application which implement required facility and provides interface so that user need not to worry about how it will work (programming). User can simply use it by interfacing with application.

Example: - CAD, medical and business systems.

Output primitives: Line drawing Algorithm

- Algorithms for displaying straight lines are based on the line equation
  - The Cartesian slope-intercept equation for a straight line is $y = m \cdot x + b$

    m slope of the line
    b y intercept.
  - Given that the two endpoints specified at positions (x1, y1) and (x2, y2, we can determine values for the slope m and y intercept b with:

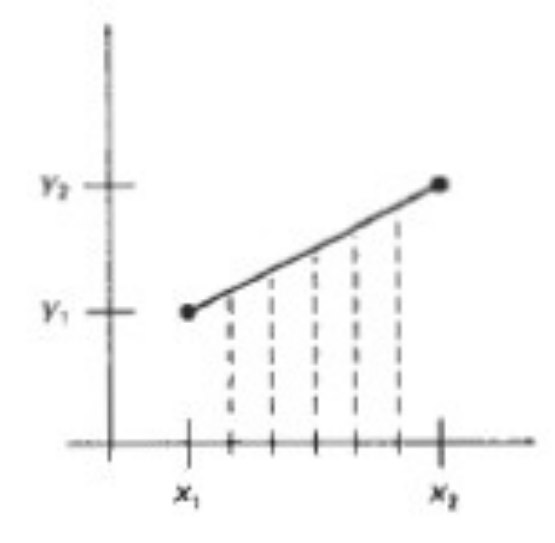
    $m = \frac{y2 - y1}{x2 - x1}$

    b=y1-m.x1
  - For any given x interval Δx along a line, corresponding y interval Δy $\Delta y = m \, \Delta x$: Similarly, x interval Δx corresponding to a specified Δy $\Delta x = \frac{\Delta y}{m}$

    These equations form the basis for determining deflection voltages in analog devices.
  - For lines with slope magnitudes, |m| < 1, Δx can be set proportional to a small horizontal deflection voltage and the corresponding vertical deflection is set proportional to Δy.
  - For lines with slopes have magnitudes |m|> 1, Δy can be set proportional to a small vertical deflection voltage with the corresponding horizontal deflection voltage set proportional to Δx.
  - For lines with m = 1, Δx = Δy and the horizontal and vertical deflections voltages are equal.

Output primitives: Line drawing Algorithm

- In fact case, a smooth line with slope m is generated between the specified endpoints.
- On raster systems, lines are plotted with pixels, and step sizes in the horizontal and vertical directions are constrained by pixel separations.
- That is, we must "sample" a line at discrete positions and determine the nearest pixel to the line at each sampled position.
- This scan conversion process for straight lines for a near horizontal line with discrete sample positions along the x axis.



Output primitives: Line drawing Algorithm- Digital Differential Analyser(DDA)

- 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 five sampling positions along line path for the other coordinate.
- Consider first a line with positive slope.
- If the slope is less than or equal to 1, we sample at unit x intervals ($\Delta x = 1$) and compute each successive y value as $y_{k+1} = y_k + m$
- 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.
- For lines with a positive slope greater than 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 $x_{k+1} = x_k + \frac{1}{m}$
- If this processing is reversed, so that the starting endpoint is at the right, then either we have $\Delta x = -1$ and $y_{k+1} = y_k - m$ or when the slope is greater than 1, $\Delta y = -1$ with $x_{k+1} = x_k - \frac{1}{m}$
- The above equations used to calculate pixel positions along a line with negative slope.

Output primitives: Line drawing Algorithm- Digital Differential Analyser(DDA)

- If the absolute value of the slope is less than 1 and the start endpoint is at the left, we set $\Delta x = 1$ and calculate y values.
- When the start endpoint is at the right (for the same slope), we set $\Delta x = -1$ and obtain y positions.
- Similarly, when the absolute value of a negative slope is greater than 1, we use $\Delta y = -1$ and use $\Delta y = 1$.
- Horizontal and vertical differences between the endpoint positions are assigned to parameters dx and dy.
- The difference with the greater magnitude determines the value of parameter steps.
- Starting with pixel position (xa, ya), we determine the offset needed at each step to generate the next pixel position along the line path.
- We loop through this process steps times. If the magnitude of dx is greater than the magnitude of dy and xa is less than xb, the values of the increments in the x and y directions are 1 and m, respectively.
- If the greater change is in the x direction, but xa is greater than xb, then the decrements - 1 and -m are used to generate each new point on the line.
- Otherwise, we use a unit increment (or decrement) in they direction and an x increment (or decrement) of l / m .
- The DDA algorithm is a faster method for calculating pixel positions.
- It eliminates the multiplication and use of raster characteristics, so that appropriate increments are applied in the x or y direction to step to pixel positions along line path.

Output primitives: Line drawing Algorithm- Digital Differential Analyser(DDA)

- The accumulation of roundoff error in successive additions of the floating-point increment, however, can cause the calculated pixel positions to drift away from the true line path for long line segments.
- Furthermore, the rounding operations and floating-point arithmetic in procedure lineDDA are still time-consuming.
- We can improve the performance of DDA by separating the increments m and 1/m into integer and fractional parts so that all calculations are reduced to integer operations.
- A method for calculating 1m increments in integer steps.

Output primitives: Line drawing Algorithm- Digital Differential Analyse

ALGORITHM:

```
procedure  line DDA (xa,ya,xb,yb:integer)
var
dx,dy,steps,K:integer;
xIncrement,yIncrement,x,y:real;
begin
      dx=xb-xa;
      dy=yb-ya;
      if abs(dx)>abs(dy) then steps = abs(dx)
      else steps = abs(dy);
      xIncrement = dx/steps;
      yIncrement = dy/steps;
      setpixel(round(x),round(y),1);

 for K=1 to steps do
   begin
         x=x+xIncrement;
         y=y+yIncrement;
       setpixel(round(x),round(y),1);
   end
end;(line DDA)
```

Output primitives: Line drawing Algorithm- Digital Differential Analyser(DDA)

```
#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));
  }
}
```

Output primitives: Line drawing Algorithm- Digital Differential Analyser(DDA)

EXAMPLE PROBLEM:

1) (2,2) to (10,3)

dx = x2-x1 → (10-2)=8
dy = y2-y1 → (3-2)=1

dx>dy (8>1)

steps=8

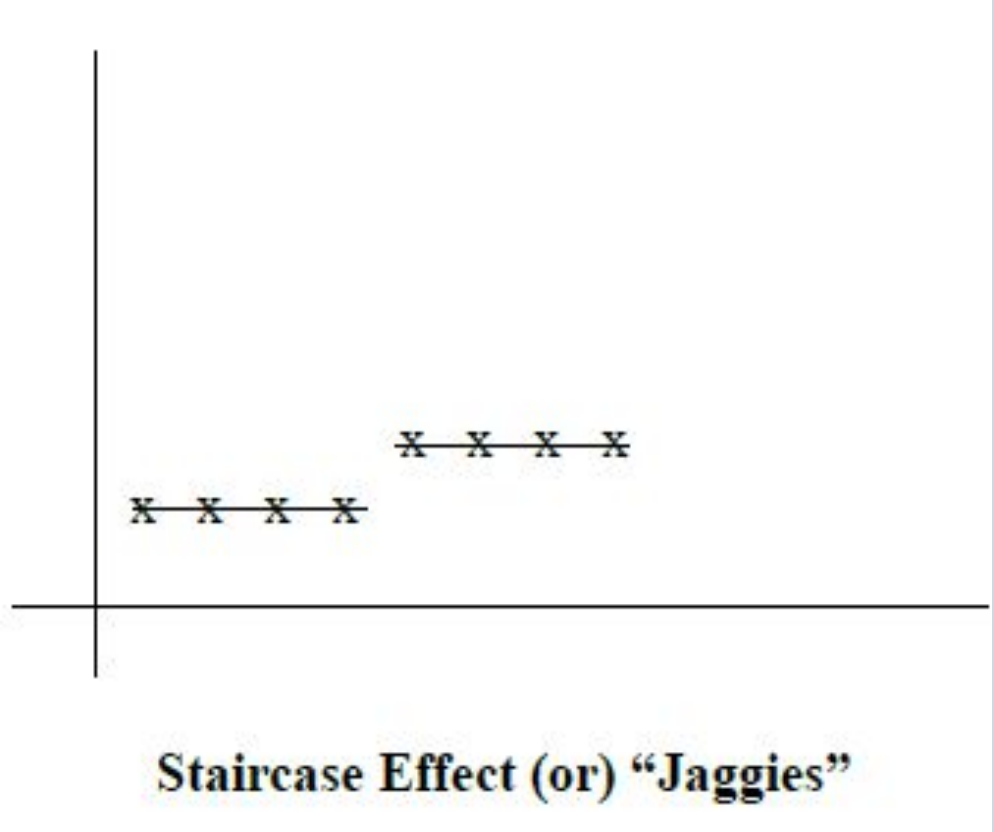
xIncrement = dx/steps → 8/8 = 1
yIncrdement = dy/steps → 1/8

x = x1 → 2
y = y1 → 2
setpixel(round(2),round(2),1); → (2,2)

for K=1 to 8 do

K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
x=2+1=3 y=2+(1/8) =2.215	x=3+1=4 y=2.125+(1/8) =2.25	x=4+1=5 y=2.25+(1/8) =2.375	x=5+1=6 y=2.375+(1/8) =2.5	x=6+1=7 y=2.5+(1/8) =2.625	x=7+1=8 y=2.625+(1/8) =2.75	x=8+1=9 y=2.75+(1/8) =2.875	x=9+1=10 y=2.875+(1/8) =3
Rounding							
(3,2) setpixel(3,2,1)	(4,2)	(5,2)	(6,3)	(7,3)	(8,3)	(9,3)	(10,3)

Output primitives: Line drawing Algorithm- Digital Differential Analyser(DDA)



Staircase Effect (or) “Jaggies”

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

- An accurate and efficient raster line-generating algorithm, scan converts lines using only incremental integer calculations that can be adapted to display circles and other curves.
- Vertical axes show-scan-line positions, and the horizontal axes identify pixel columns.
- Starting from the left endpoint, we need to determine at the next sample position whether to plot the pixel at position.
- Negative slope-line path starting from the left endpoint at pixel position.
- These questions are answered with Bresenham's line algorithm by testing the sign of an integer parameter, whose value is proportional to the difference between the separations of the two pixel positions from the actual line path.
- To illustrate Bresenham's approach, consider the scan-conversion process for lines with positive slope less than 1.
- Pixel positions along a line path are then determined by sampling at unit x intervals.
- Starting from the left endpoint (xo, yo) 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 (xk, yk) is to be displayed, we next need to decide which pixel to plot in column xk+1.
- Our choices are the pixels at positions xk+1,yk and (xk+1, yk+1).
- At sampling position xk+1, we label vertical pixel separations from the mathematical line path as d, and d2.
- y coordinate on the mathematical line at pixel column position xk+1 is calculated as

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

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

Then $$d_1 = y - y_i = m(x_k + 1) + b - y_k$$

And $$d_2 = (y_k + 1) - y = y_k + 1 - m(x_k + 1) - b$$

The difference between $$d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2b - 1$$

A decision parameter pk for the kth step in the line algorithm can be obtained by rearranging so that it involves only integer calculations. We accomplish this by substituting m = Δy/ Δx, where Δy and Δx are the vertical and horizontal separations of the endpoint positions, and defining $$p_k = \Delta x(d_1 - d_2) = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$$

The sign of p, is the same as the sign of d1- d2, since Δx > 0 for our example. Parameter c is constant and has the value 2Ay + Ax(2b - l), which is independent of pixel position and will be eliminated in the recursive calculations for pk. If the pixel at yk is closer to the line path than the pixel at yk+l (that is, d1 < d2), decision parameter pk is negative. In that case, we plot the lower pixel; else plot upper pixel.

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

- Coordinate changes along the line occur in unit steps in either the x or y directions.
- Therefore, we can obtain the values of successive decision parameters using incremental integer calculations. At step k + 1, the decision parameter is evaluated

$$p_{k+1} = 2\Delta y \cdot x_{k+1} - 2\Delta x \cdot y_{k+1} + c$$

- Subtracting from the preceding equation, $p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\,\Delta x(y_{k+1} - y_k)$
- But $x_{k+1} = x_k + 1$

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

- where the term $y_{k+1} - y_k$ or 1, depending on the sign of parameter pk.
- This recursive calculation of decision parameters is performed at each integer
- x position, starting at the left coordinate endpoint of the line. The first parameter,
- p0, at the starting pixel position (xo, yo) and with m evaluated as $\Delta y/\Delta x$

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

- The constants $2\Delta y$ and $2\Delta y - 2\Delta x$ once for each line to be scan converted, so the arithmetic involves only integer addition and subtraction of these two constants.

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

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.

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

```
#include "device.h"

void lineBres (int xa, int ya, int xb, int yb)
{
  int dx = abs (xa - xb), dy = abs (ya - yb);
  int p = 2 * dy - dx;
  int twoDy = 2 * dy, twoDyDx = 2 * (dy - dx);
  int x, y, xEnd;

  /* Determine which point to use as start, which as end */
  if (xa > xb) {
    x = xb;
    y = yb;
    xEnd = xa;
  }
  else {

     x = xa;
     y = ya;
     xEnd = xb;
   }
   setPixel (x, y);

   while (x < xEnd) {
     x++;
     if (p < 0)
       p += twoDy;
     else {
       y++;
       p += twoDyDx;
     }
     setPixel (x, y);
   }
 }
```

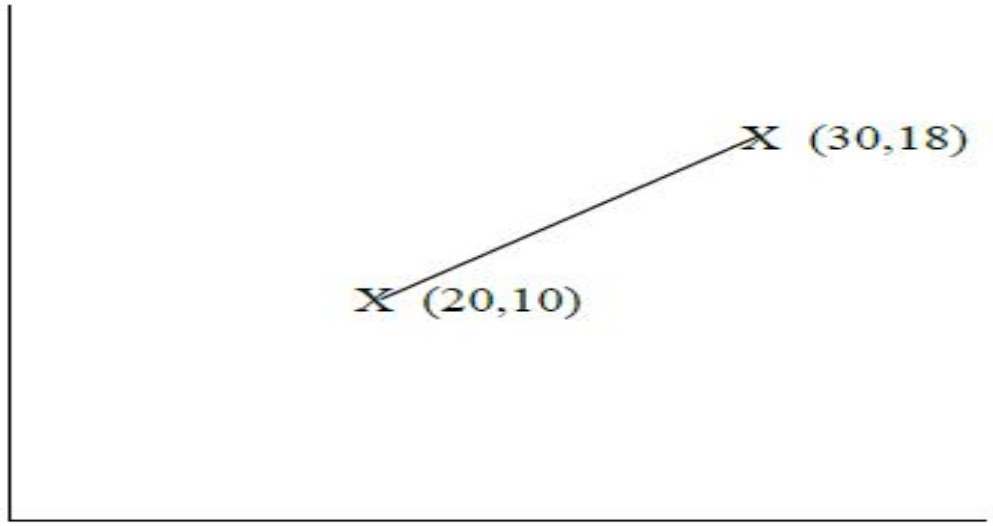
Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

- Bresenham's algorithm is generalized to lines with arbitrary slope by considering the symmetry between the various octants and quadrants of the xy plane.
- For a line with positive slope greater than 1, we interchange the roles of the x and y directions.
- That is, we step along they direction in unit steps and calculate successive x values nearest the line path.
- Also, we could revise the program to plot pixels starting from either endpoint.
- If the initial position for a line with positive slope is the right endpoint, both x and y decrease as we step from right to left.
- To ensure that the same pixels are plotted regardless of the starting endpoint, we always choose the upper (or the lower) of the two candidate pixels whenever the two vertical separations from the line path are equal (d1 = d2).
- For negative slopes, the procedures are similar, except that now one coordinate decreases as the other increases.
- Finally, specla1 cases can be handled separately:
- Horizontal lines (Δy = 0), vertical lines (Δr = O), and diagonal lines with |Δx| = |Δy| each can be loaded directly into the frame buffer without processing them through the line-plotting algorithm.

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

EXAMPLE:

Line endpoints (20,10) and (30,18). Calculate the pixel value.



Solution:

Slope = $\Delta y/\Delta x = (18\text{-}10)/(30\text{-}10) = 8/10 = 0.8$.

Slope < 1

$\Delta x = 10$, $\Delta y = 8$, $2\Delta y = 16$, $2\Delta y - 2\Delta x = -4$

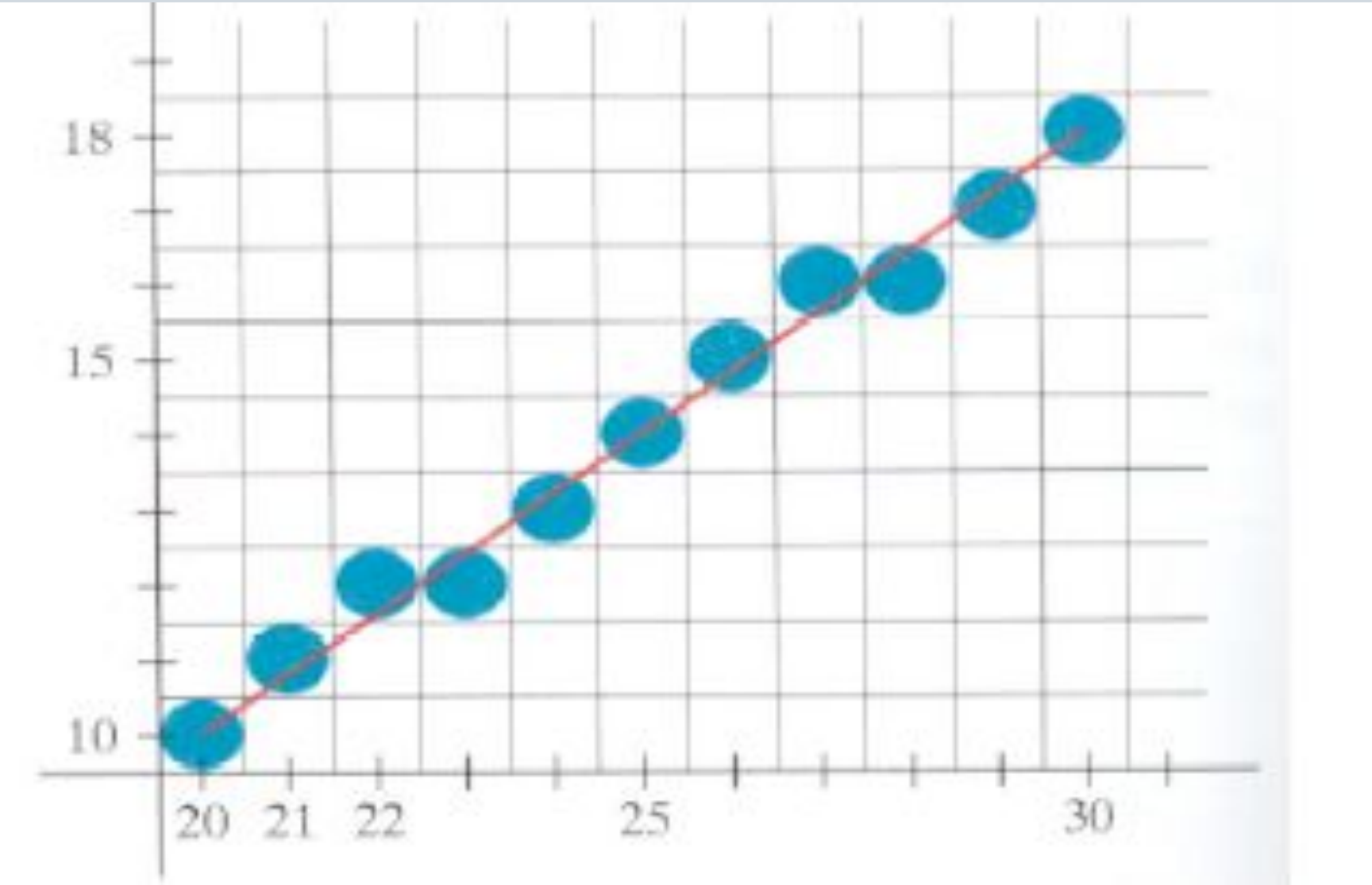
Plot the initial point $(x_0,y_0) = (20,10)$

The successive pixel positions are based on the decision parameter.

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm

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)

Output primitives: Line drawing Algorithm- Bresenham's Line Algorithm



Midpoint Circle Algorithm

- As in the raster line algorithm, we sample at unit intervals and determine the closest pixel position to the specified circle path at each step.
- For a given radius r and screen center position (xc , yc), we can first set up our algorithm to calculate pixel positions around a circle path centered at the coordinate origin (0,O).
- Each calculated position (x, y) is moved to its proper screen position by adding xc, to x and yc 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.
- Therefore, we can take unit steps in the positive x direction over this octant and use a decision parameter to determine which of the two possible y positions is closer to the circle path at each step.
- Positions in the other seven octants are then obtained by symmetry.
- To apply the midpoint method, we define a circle function: $f_{circle}(x, y) = x^2 + y^2 - r^2$

- Any point (x , y) on the boundary of the circle with radius r satisfies fcircle(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.
- Relative position of any point (x. v) can be determined by checking sign of circle function:

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

- The circle-function tests are performed for the mid positions between pixels near the circle path at each sampling step.

Midpoint Circle Algorithm

- Thus, 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 (xk, yk), we next need to determine whether the pixel at position (xk+ 1, yk) or the one at position (xk + 1, yk - 1) is closer to the circle.
- Decision parameter is circle function evaluated at the midpoint between these two pixels:

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

- If pk < 0, this midpoint is inside the circle and the pixel on scan line yb is closer to the circle boundary.
- Otherwise, the mid-position is outside or on the circle boundary, and we select the pixel on
  scanline yk - 1.
- Successive decision parameters are obtained using incremental calculations.
- We obtain a recursive expression for the next de
  circle function at sampling position xk+1 + 1 = x

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

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

- where yk+1 is either yk or yk-1, depending on the sign of pk.

Midpoint Circle Algorithm

- Increments for obtaining pk+1, are either 2xk+1 + 1 (if pk is negative) or 2xk+1+1.
- Evaluation of the terms 2xk+1 and 2yk+1 can also be done incrementally as

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

- At the start position (0, T), these two terms have the values 0 and 2r, respectively.
- Each successive value is obtained by adding 2 to the previous value of 2x and subtracting 2 from the previous value of 2y.
- The initial decision parameter is obtained by evaluating the circle function at the start position (x0, yo) = (0, r):

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

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

- If the radius r is specified as an integer, we can simply round po to

$$p_0 = 1 - r \qquad \text{(for } r \text{ an integer)}$$

- since all increments are integers.
- As in Bresenham's line algorithm, the midpoint method calculates pixel positions along the circumference of a circle using integer additions and subtractions, assuming that the circle parameters are specified in integer screen coordinates.

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 Algorithm

Mid-Point Circle Algorithm

Example:

Given a circle of radius r=10. Demonstrate the midpoint circle algorithm by determining the positions along the circle octant in first quadrant from x=0 to x=y.

$P_0 = 1-r = 1-10 = -9$

For a circle centered on coordinate orgin, the initial point is $(x_0,y_0)=(0,10)$

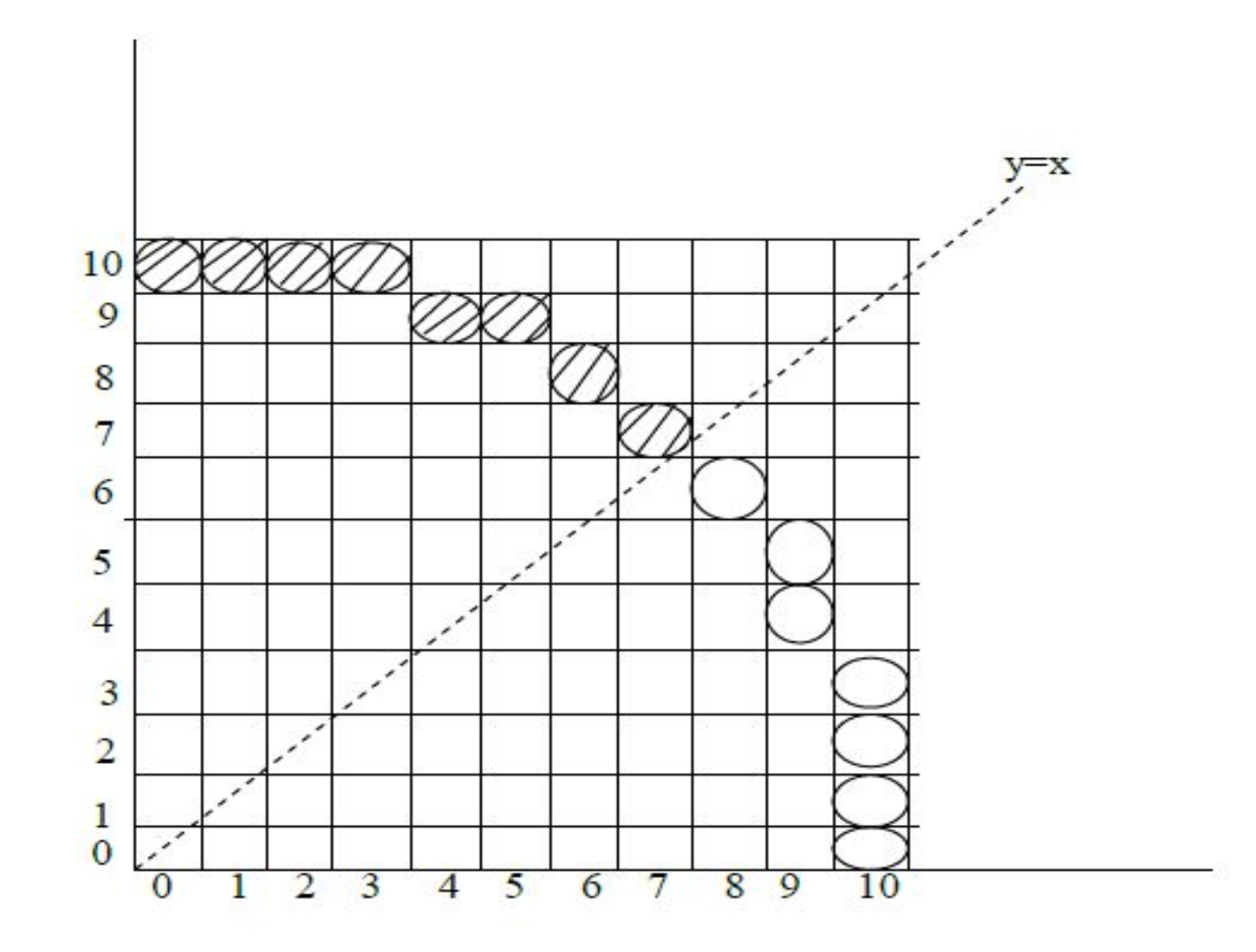
Initial increment terms for calculating decision parameters,

$2x_0 = 0 \ , \ 2y_0 = 20$

Successive decision parameter values & position along circle path.

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

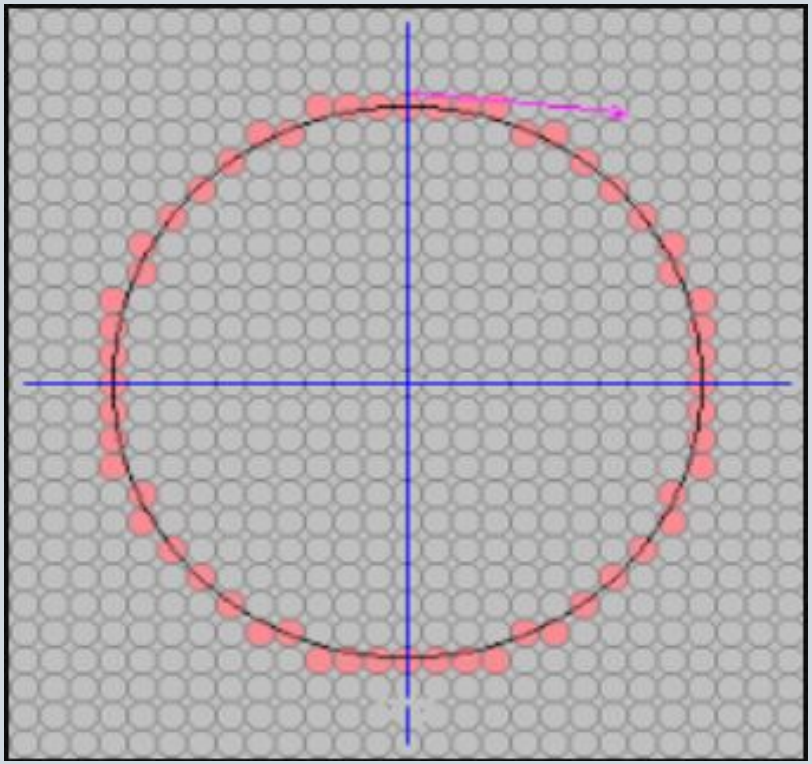
Mid-Point Circle Algorithm



Mid-Point Circle Algorithm

Quadrant-1 (X,Y)	Quadrant-2 (-X,Y)	Quadrant-3 (-X,-Y)	Quadrant-4 (X,-Y)
(0, 10)	(0, 10)	(0, -10)	(0, -10)
(1, 10)	(-1, 10)	(-1, -10)	(1, -10)
(2, 10)	(-2, 10)	(-2, -10)	(2, -10)
(3, 10)	(-3, 10)	(-3, -10)	(3, -10)
(4, 9)	(-4, 9)	(-4, -9)	(4, -9)
(5, 9)	(-5, 9)	(-5, -9)	(5, -9)
(6, 8)	(-6, 8)	(-6, -8)	(6, -8)
(8, 6)	(-8, 6)	(-8, -6)	(8, -6)
(9, 5)	(-9, 5)	(-9, -5)	(9, -5)
(9, 4)	(-9, 4)	(-9, -4)	(9, -4)
(10, 3)	(-10, 3)	(-10, -3)	(10, -3)
(10, 2)	(-10, 2)	(-10, -2)	(10, -2)
(10, 1)	(-10, 1)	(-10, -1)	(10, -1)
(10, 0)	(-10, 0)	(-10, 0)	(10, 0)
These are all points of the Circle.			

Mid-Point Circle Algorithm



Ellipse Generating Algorithm

Ellipse Generating Algorithms

- Equation of ellipse:

$$d_1 + d_2 = constant$$

- F1→(x_1,y_1), F2→(x_2,y_2)

$$\sqrt{(x-x_1)^2+(y-y_1)^2}+\sqrt{(x-x_2)^2+(y-y_2)^2}=constant$$

- General Equation

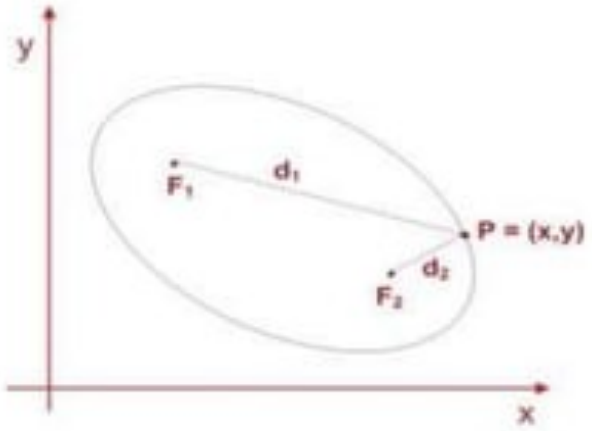
$$Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$$

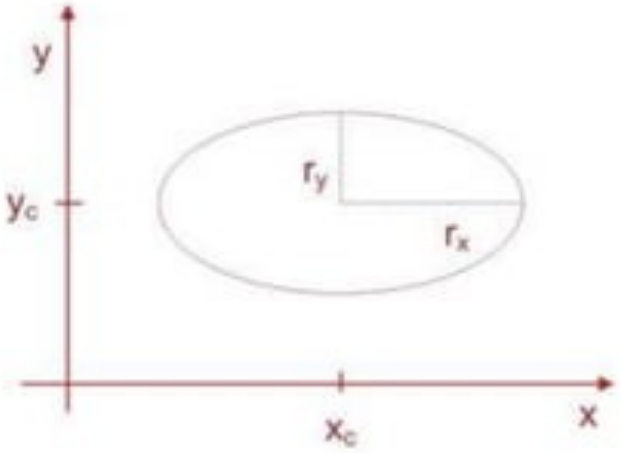
- Simplified Form

$$\left(\frac{x-x_c}{r_x}\right)^2+\left(\frac{y-y_c}{r_y}\right)^2=1$$

- In polar co-ordinate

$$x = x_c + r_x \cos\theta$$
$$y = y_c + r_y \sin\theta$$





Ellipse Generating Algorithm

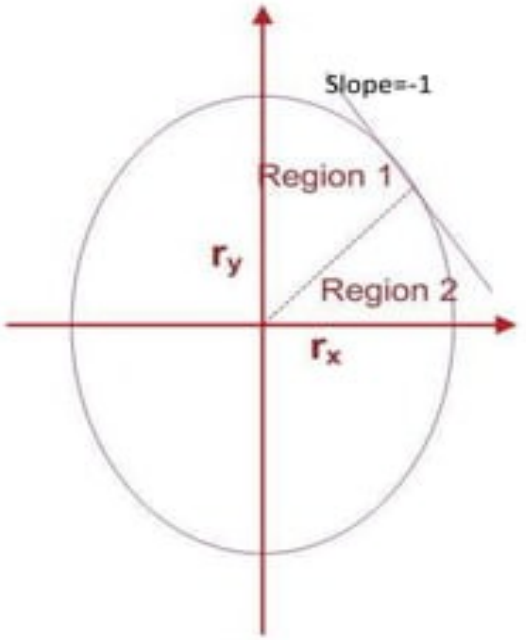
- Ellipse function:

$$f_{ellipse}(x,y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$$

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

- From ellipse tangent slope: $\frac{dy}{dx} = -\frac{2r_y^2 x}{2r_x^2 y}$
- At boundary region (slope = -1): $2r_y^2 x = 2r_x^2 y$
- Start from $(0, r_y)$, take x samples to boundary between 1 and 2
- Switch to sample y from boundary between 1 and 2

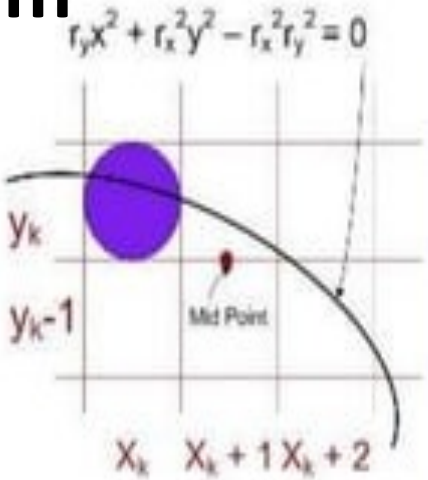
  (i.e whenever $2r_y^2 x \geq 2r_x^2 y$)



Ellipse Generating Algorithm

- In the region 1

$$p1_k = f_{ellipse}(x_k + 1, y_k - \frac{1}{2})$$
$$= r_y^2 (x_k + 1)^2 + r_x^2 (y_k - \frac{1}{2})^2 - r_x^2 r_y^2$$



- For next sample

$$p1_{k+1} = f_{ellipse}(x_{k+1} + 1, y_{k+1} - \frac{1}{2})$$
$$= r_y^2 [(x_k + 1) + 1]^2 + r_x^2 (y_{k+1} - \frac{1}{2})^2 - r_x^2 r_y^2$$
$$p1_{k+1} = p1_k + 2r_y^2 (x_k + 1) + r_y^2 + r_x^2 \left[\left(y_{k+1} - \frac{1}{2} \right)^2 - \left(y_k - \frac{1}{2} \right)^2 \right]$$

- Thus increment

$$increment = \begin{cases} 2r_y^2 x_{k+1} + r_y^2, & if \quad p1_k < 0 \\ 2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1}, & if \quad p1_k \geq 0 \end{cases}$$

- For increment calculation; Initially:

$$2r_y^2 x = 0$$
$$2r_x^2 y = 2r_x^2 r_y$$

- Incrementally:
  Update x by adding $2r_y^2$ to first equation and update y by subtracting $2r_x^2$ to second equation

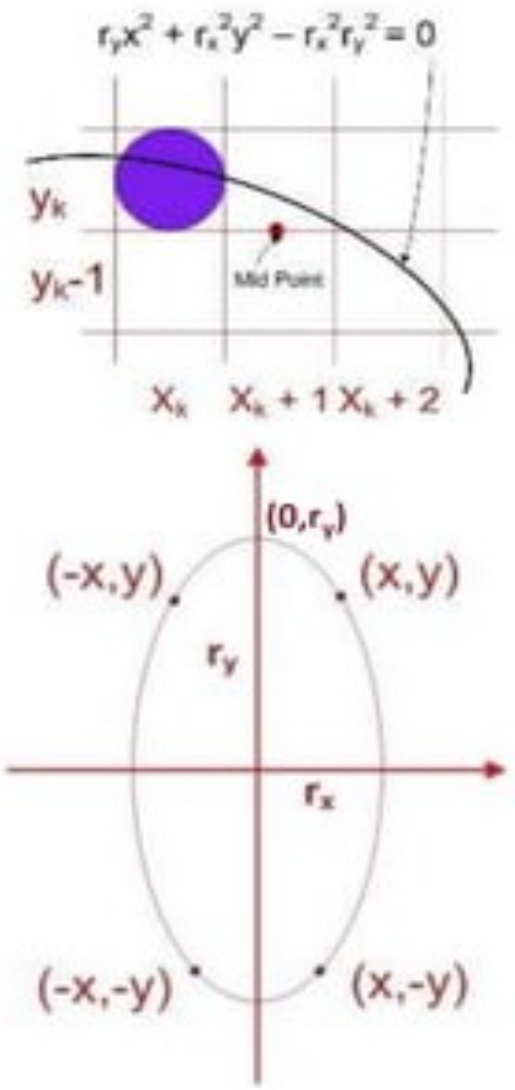
Ellipse Generating Algorithm

- Initial value

$$p1_0 = f_{ellipse}\left(1, r_y - \frac{1}{2}\right)$$

$$= r_y^2 + r_x^2\left(r_y - \frac{1}{2}\right)^2 - r_x^2 r_y^2$$

$$p1_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$



Ellipse Generating Algorithm

- In the region 2

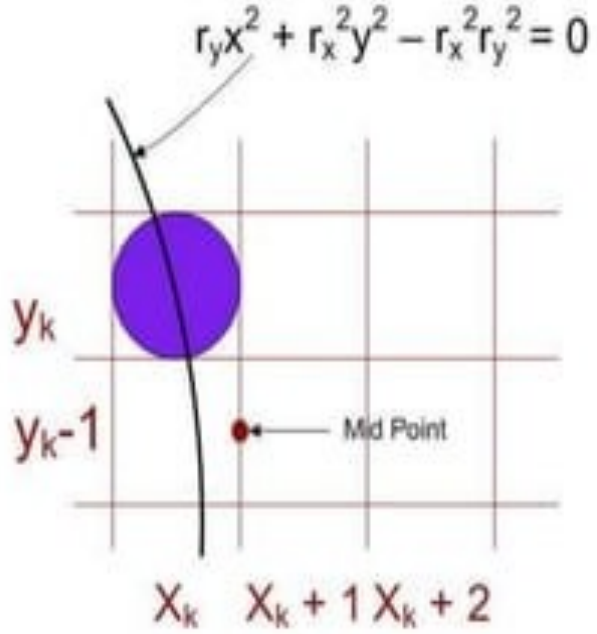
$$p2_k = f_{ellipse}(x_k + \frac{1}{2}, y_k - 1)$$
$$= r_y^2 (x_k + \frac{1}{2})^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2$$

- For next sample

$$p2_{k+1} = f_{ellipse}(x_{k+1} + \frac{1}{2}, y_{k+1} - 1)$$
$$= r_y^2 \left(x_{k+1} + \frac{1}{2} \right)^2 + r_x^2 [(y_k - 1) - 1]^2 - r_x^2 r_y^2$$
$$p2_{k+1} = p2_k + 2r_x^2 (y_k - 1) + r_x^2 + r_y^2 \left[\left(x_{k+1} + \frac{1}{2} \right)^2 - \left(x_k - \frac{1}{2} \right)^2 \right]$$

- Initially

$$p2_0 = f_{ellipse}\left(x_0 + \frac{1}{2}, y_0 - 1 \right)$$
$$p2_0 = r_y^2 \left(x_0 + \frac{1}{2} \right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$



For simplification calculation of $p2_0$ can be done by selecting pixel positions in counter clockwise order starting at $(r_x, 0)$ and unit samples to positive y direction until the boundary between two regions

Ellipse Generating Algorithm

Algorithm

1. Input r_x,r_y and the ellipse center(x_c,yc) and obtain the first point on an ellipse centered on the origin as
   $(x_0,y_0) = (0,r_y)$
2. Calculate the initial value of the decision parameter in region 1 as

$$p1_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$

3. At each x_k position in region 1, starting at k = 0, perform the following test: If $p1_k < 0$, the next point along the ellipse centered on (0,0) is (x_{k+1},y_k) and

$$p1_{k+1} = p1_k + 2r_y^2 x_{k+1} + r_y^2$$

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

With $$p1_{k+1} = p1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$$

$$2r_y^2 x_{k+1} = 2r_y^2 x_k + 2r_y^2, \qquad 2r_x^2 y_{k+1} = 2r_x^2 y_k - 2r_x^2$$

and continue until

$$2r_y^2 x \geq 2r_x^2 y$$

Ellipse Generating Algorithm

4. Calculate the initial value of decision parameter in region 2 using the last point (x_0,y_0) calculated in region 1 as

$$p2_0 = r_y^2\left(x_0+\frac{1}{2}\right)^2 + r_x^2(y_0-1)^2 - r_x^2 r_y^2$$

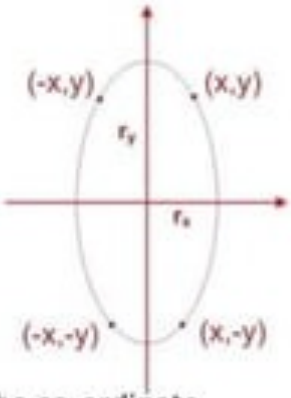
5. At each y_k position in region 2, starting at k = 0, perform the following test: If p2k>0, the next point along the ellipse centered on (0,0) is (x_k,y_k-1) and

$$p2_{k+1} = p2_k - 2r_x^2 y_{k+1} + r_x^2$$

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

$$p2_{k+1} = p2_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$$

Using the same incremental calculations for x and y as in region 1.



6. Determine the symmetry points in the other three quadrants.
7. Move each calculated pixel position (x,y) onto the elliptical path centered on (x_c,y_c) and plot the co-ordinate values:

$X = x + x_c$, $y = y + y_c$

8. Repeat the steps for region 1 until

$$2r_y^2 x \geq 2r_x^2 y$$

Ellipse Generating Algorithm

- Digitize the ellipse with parameter $r_x=8$ and $r_y=6$ in first quadrant.

Solution:

$2r_y^2x = 0$ (with increment $2r_y^2 = 72$)

$2r_x^2y = 2r_x^2r_y$ (with increment $-2r_x^2 = -128$)

$$p1_0 = r_y^2 - r_x^2r_y + \frac{1}{4}r_x^2 = -332$$

Ellipse Generating Algorithm

k	$p1_k$	(x_{k+1}, y_{k+1})	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	-332	(1, 6)	72	768
1	-224	(2, 6)	144	768
2	-44	(3, 6)	216	768
3	208	(4, 5)	288	640
4	-108	(5, 5)	360	640
5	288	(6, 4)	432	512
6	244	(7, 3)	504	384

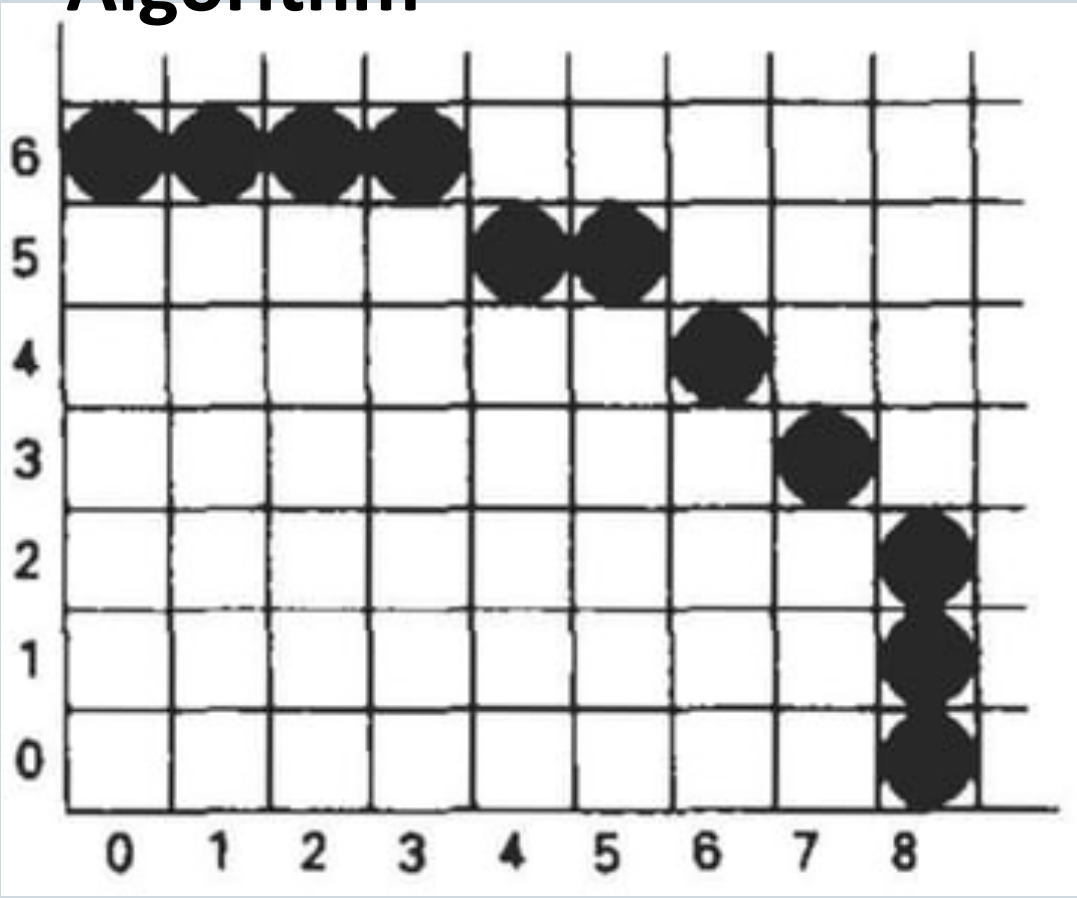
Ellipse Generating Algorithm

For region 2, the initial point is $(x_0, y_0) = (7, 3)$ and the initial decision parameter is

$$p2_0 = f\left(7 + \frac{1}{2}, 2\right) = -151$$

k	$p2_k$	(x_{k+1}, y_{k+1})	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
0	−151	(8, 2)	576	256
1	233	(8, 1)	576	128
2	745	(8, 0)	—	—

Ellipse Generating Algorithm



Pixel addressing

- A screen coordinate position is the pair of integer values identifying a grid intersection position between two pixels.
- With the coordinate origin at the lower left of the screen, each pixel area can be referenced by the integer grid coordinates of its lower left corner.
- In general, we identify the area occupied by a pixel with screen coordinates (x, y) as the unit square with diagonally opposite corners at (x, y) and (x + 1, y + 1).

This pixel-addressing scheme advantages:

- It avoids half-integer pixel boundaries, it facilitates precise object representations, and it simplifies the processing involved in many scan-conversion algorithms and in other raster procedures.
- The algorithms for line drawing and curve generation discussed in the preceding sections are still valid when applied to input positions expressed as screen grid coordinates.
- Decision parameters in these algorithms are now simply a measure of screen grid separation differences, rather than separation differences from pixel centres.
- For an enclosed area, input geometric properties are maintained by displaying the area only with those pixels that are interior to the object boundaries.

Filled area primitives:

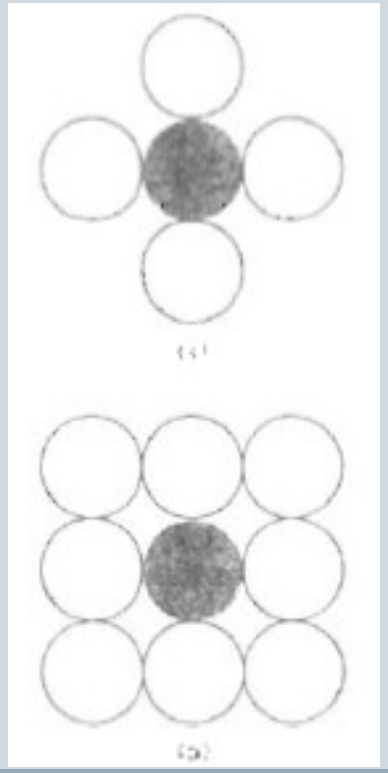
- A standard output primitive in general graphics packages is a solid-color or patterned polygon area.
- Other kinds of area primitives are sometimes available, but polygons are easier to process since they have linear boundaries
- There are two basic approaches to area filling on raster systems.
- One way to fill an area is to determine the overlap intervals for scan lines that cross the area.
- Another method for area filling is to start from a given interior position and paint
- outward from this point until we encounter the specified boundary conditions.
- The scan-line approach is typically used in general graphics packages to fill polygons, circles, ellipses, and other simple curves.
- All methods starting from an interior point are useful with more complex boundaries and in interactive painting systems.

Filled area primitives: Boundary Fill algorithms

- To fill area is to start at a point inside a region and paint the interior outward toward the boundary.
- If the boundary is specified in a single color, the fill algorithm proceeds outward pixel by pixel until the boundary color is encountered. This method, called the boundary-till algorithm.
- It is particularly useful in interactive painting packages, where interior points are easiiy selected.
- Using a graphics tablet or other interactive device, an artist or designer can sketch a figure outline, select a fill color or pattern from a color menu, and pick an interior point.
- The system then paints the figure interior.
- To display a solid color region (with no border), the designer can choose the fill color to be the same as the boundary color.
- A boundary-fill procedure accepts as input the coordinates of an interior point (x, y), a fill color, and a boundary color.
- Starting from (x, y), the procedure tests neighboring positions to determine whether they are of the boundary color.
- If not, they are painted with the fill color, and their neighbors are tested.
- This process continues until all pixels up to the boundary color for the area have been tested.
- Both inner and outer boundaries can be set up to specify an area.
- These are the pixel positions that are right, left, above, and below the current pixel.
- Areas filled by this method are called 4-connected.
- The second method, shown in
- Fig. 3-43(b), is used to fill more complex figures. Here the set of neighboring positions
- to be tested includes the four diagonal pixels. Fill methods using this approach

Filled area primitives: Boundary Fill algorithms

- The second method is used to fill more complex figures.
- Here the set of neighboring positions to be tested includes the four diagonal pixels.
- Fill methods using this approach are called 8-connected.
- An 8 connected boundary-fill algorithm would correctly fill the interior of the area defined, but a 4-connected boundary- fill algorithm produces the partial fill shown.



Filled area primitives: Boundary Fill algorithms

- The following procedure illustrates a recursive method for filling a 4-connected area with an intensity specified in parameter fill up to a boundary color specified with parameter boundary.
- We can extend this procedure to fill an 8 connected region by including four additional statements to test diagonal positions, such is (x + 1, y + 1).

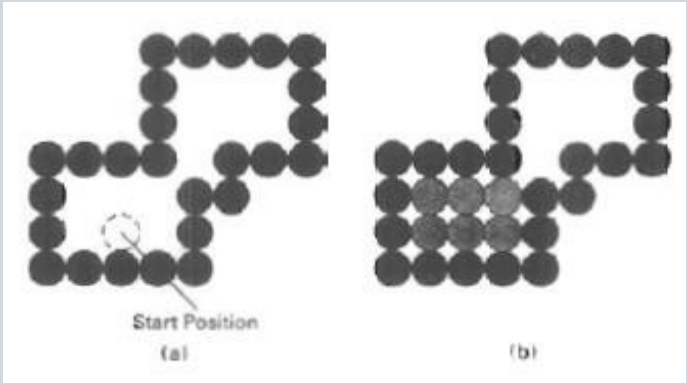
```
void boundaryFill4 (int x, int y, int fill, int boundary)
{
  int current;

  current = getPixel (x, y);
  if ((current != boundary) && (current != fill)) {
    setColor (fill);
    setPixel (x, y);
    boundaryFill4 (x+1, y, fill, boundary);
    boundaryFill4 (x-1, y, fill, boundary);
    boundaryFill4 (x, y+1, fill, boundary);
    boundaryFill4 (x, y-1, fill, boundary);
  }
}
```

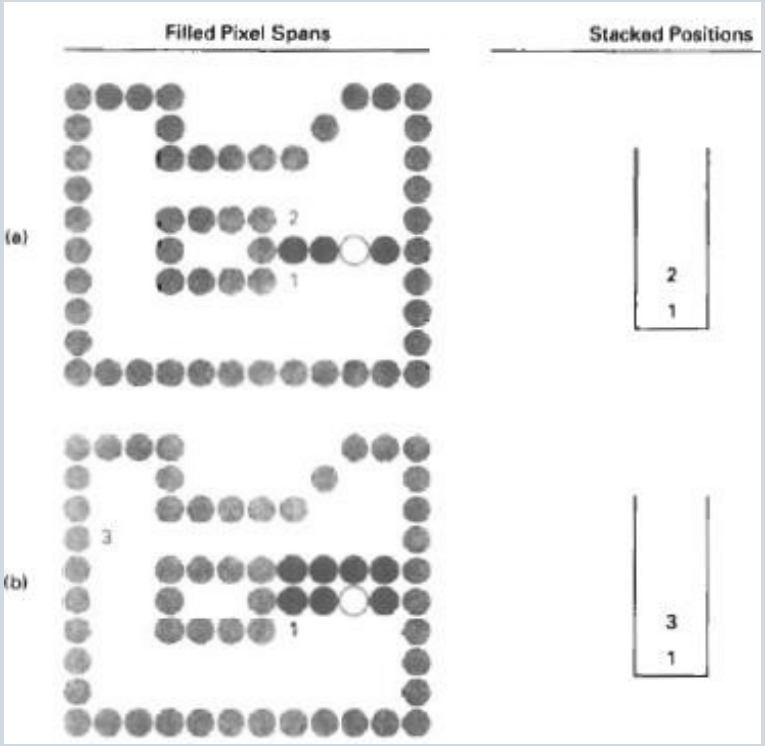
- Recursive boundary-fill algorithms may not fill regions correctly if some interior pixels are already displayed in the fill color.
- This occurs because the algorithm checks next pixels both for boundary color and for fill color.
- Encountering a pixel with the fill color can cause a recursive branch to terminate, leaving other interior pixels unfilled.
- To avoid this, we can first change the color of any interior pixels that are initially set to the fill color before applying the boundary-fill procedure.

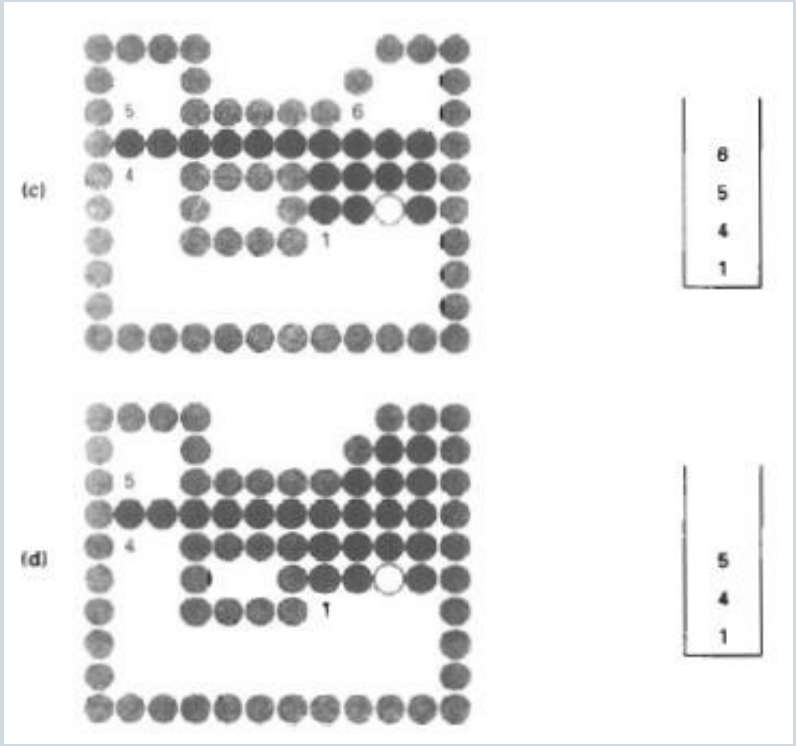
Filled area primitives: Boundary Fill algorithms

- Also, since this procedure requires considerable stacking of neighboring points, more efficient methods are generally employed.
- These methods fill horizontal pixel spans across scan lines, instead of proceeding to 4-connected or 8-connected neighboring points.
- Then we need only stack a beginning position for each horizontal pixel span, instead of stacking all unprocessed neighboring positions around the current position.
- Starting from the initial interior point with this method, we first fill in the contiguous span of pixels on this starting scan line.
- Then we locate and stack starting positions for spans on the adjacent scan lines, whew spans are defined as the contiguous horizontal string of positions bounded by pixels displayed in the area border color.
- At each subsequent step, Output Primitives we unstack the next start position and repeat the process.



Filled area primitives: Boundary Fill algorithms





Filled area primitives: Flood fill algorithms

- Sometimes we want to fill in (or recolor) an area that is not defined within a single color boundary.
- We can paint such areas by replacing a specified interior color instead of searching for a boundary color value. This approach is called a flood-fill algorithm.
- We start from a specified interior point (x, y) and reassign all pixel values that are currently set to a given interior color with the desired fill color.
- If the area we want to paint has more than one interior color, we can first reassign pixel values so that all interior points have the same color.
- Using either a 4 connected or 8-connected approach, we then step through pixel positions until all interior points have been repainted.
- We can modify procedure floodFill4 to reduce the storage requirements of the stack by filling horizontal pixel spans.
- In this approach, we stack only the beginning positions for those pixel spans having the value oldcolor .
- Starting at the first position of each span, the pixel values are replaced until a value other than oldcolor is encountered.

```
void floodFill4 (int x, int y, int fillColor, int oldColor)
{
  if (getPixel (x, y) == oldColor) {
    setColor (fillColor);
    setPixel (x, y);
    floodFill4 (x+1, y, fillColor, oldColor);
    floodFill4 (x-1, y, fillColor, oldColor);
    floodFill4 (x, y+1, fillColor, oldColor);
    floodFill4 (x, y-1, fillColor, oldColor);
  }
}
```

Attributes of Output Primitives

Structure :

- Definition
- Line Attribute
- Curve Attribute
- Color and Grayscale Level
- Area Filled Attribute
- Text and Characters

Introduction

The way a primitive is to be displayed is referred to as an *Attribute Parameter.*

Some attribute parameters include color ,size etc.

Different ways to incorporate attribute changes :

- Extend the parameter list associated with each primitive
- Maintain a system list of current attribute values and use separate functions to set attributes.

Attributes of Line

. Type

. Width

. Color

. Pen & Brush

Attributes of Line

Type :

- Solid
- Dotted – very short dash with spacing equal to or greater than dash itself
- Dashed – displayed by generating an inter dash spacing
- Dash Dotted –combination of the earlier two

To set line type attribute in PHIGS Application,

setLineType(lt) can be used. Lt can be -1,2,3,4

Attributes of Line

Raster Line algorithms display line attributes by plotting pixel spans.

Pixel count for the span and inter span length and inter span spacing can be specified using the mask .

Ex. 1111000111100011 1

Plotting dashes with fixed number of pixels result in unequal-length dashes for different line orientations. Horizontal line looks small when compared to a vertical line.

For dash lines to remain constant, we should adjust the pix span el count for the solid span and inter space span according to the slope of the line.

Line Width

Specify in pixels and proportion of a standard line width.

Thicker line can be produced by.

. Adding extra pixel vertically when $|m| < 1$

. Adding extra pixel horizontally when $|m| > 1$

We can set the width of a line using setLineWidthScaleFactor(lw);
Where lw is assigned a positive number.

Line Width

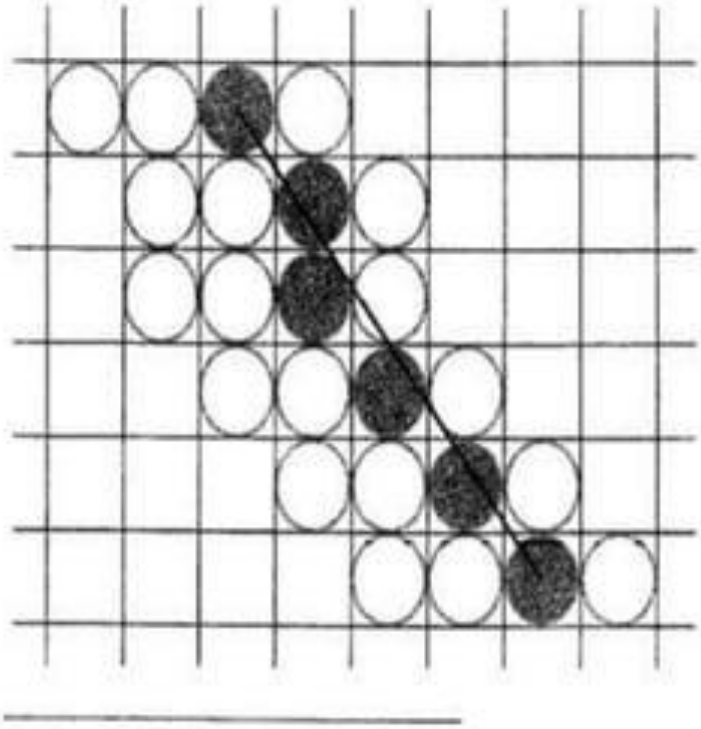
Issues:

Line have different thickness on the slope

Problem with

. End of the line (Use Line Caps)

. Joining the two lines (polygon)



Raster line with slope $|m| > 1$
and line-width parameter $l_W = 4$
plotted with horizontal pixel spans.

Line Width

Line Endcaps

Butt Cap

Round Cap

Projecting
Square Cap

Line Width

Line Joins

Miter Join

Round Join

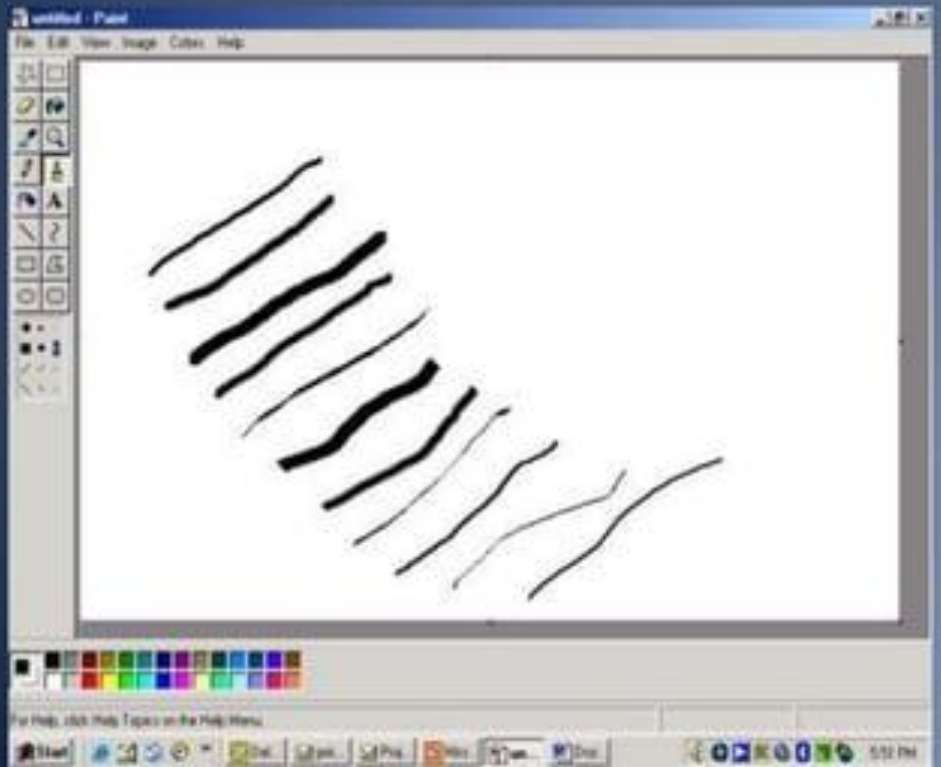
Bevel Join

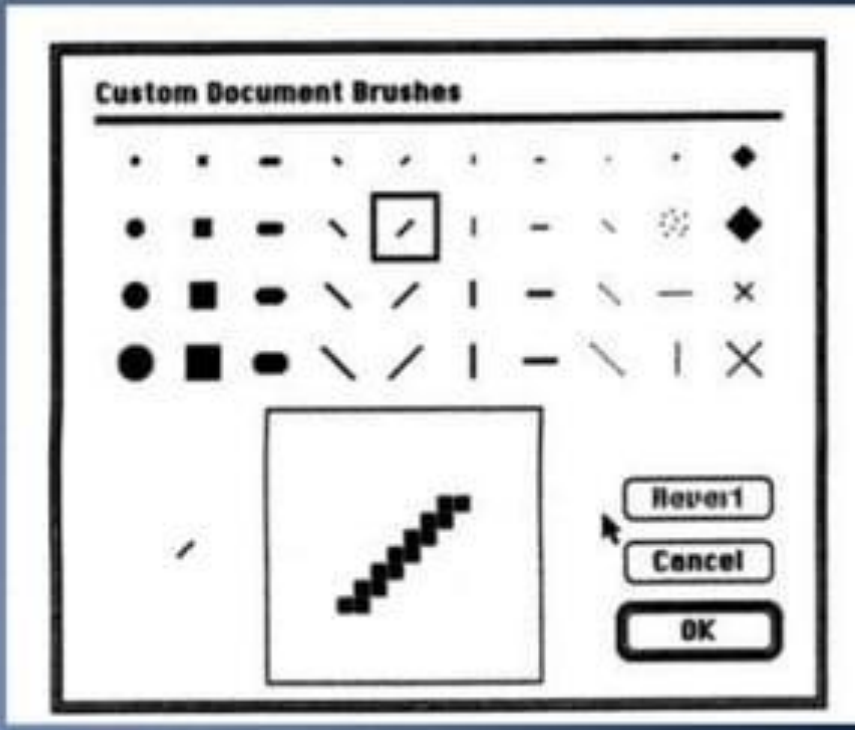
Pen and Brush Options

The selected "pen" or "brush" determine the way a line will be drawn.

Pens and brushes have size, shape, color and pattern attribute.

Pixel mask is applied in both of them.





Line Color

A polyline procedure displays a line in current color by setting this color value in the frame buffer at pixel locations along the line path using the set Pixel function.
setPolyLineColorIndex(lc)

A line drawn with background color in invisible.

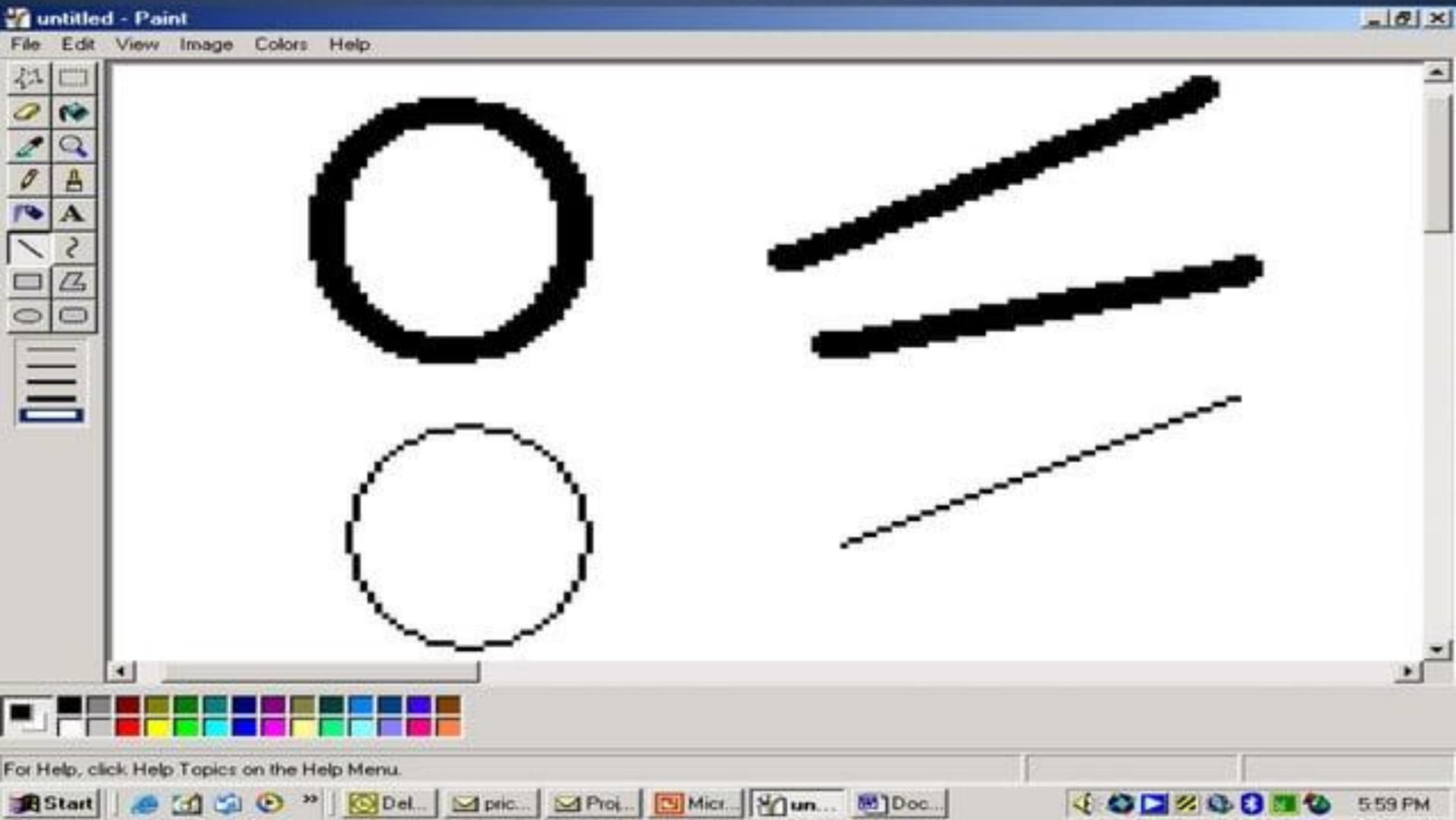
Curve Attributes

Similar to line : type + width

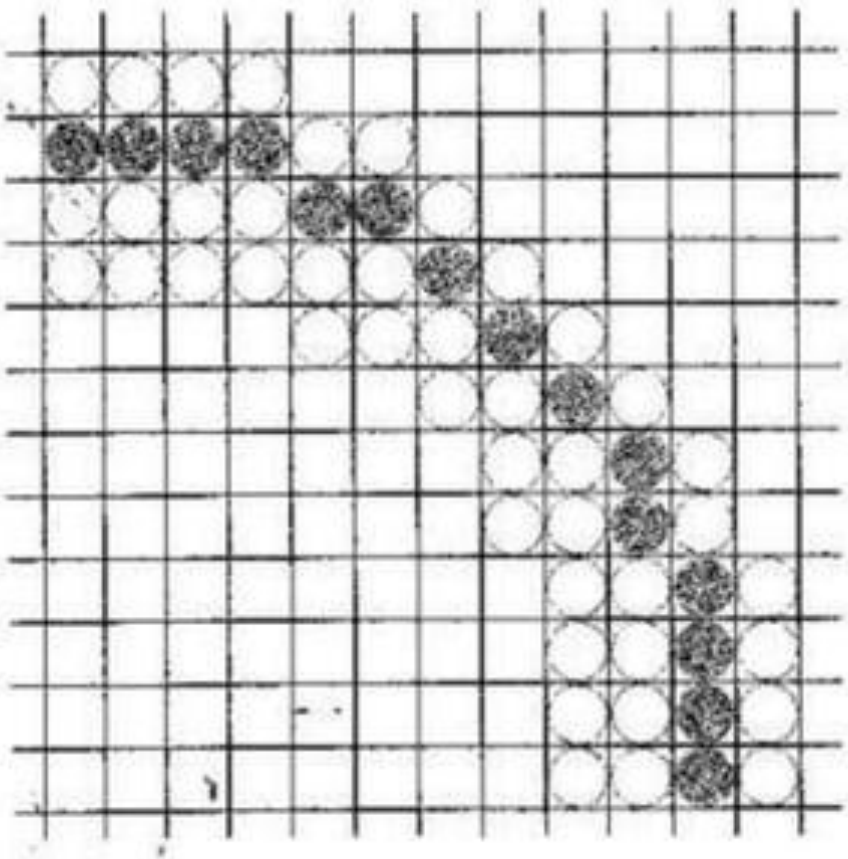
Thicker curves can be produced by:

1. Plotting additional pixel
2. Filling the space between two concentric circles.
3. Using thicker pen or brush

Curve Attributes



Curve Attributes



Circular arc of width 4 plotted with pixel spans.

Color and GreyScale Levels

Color

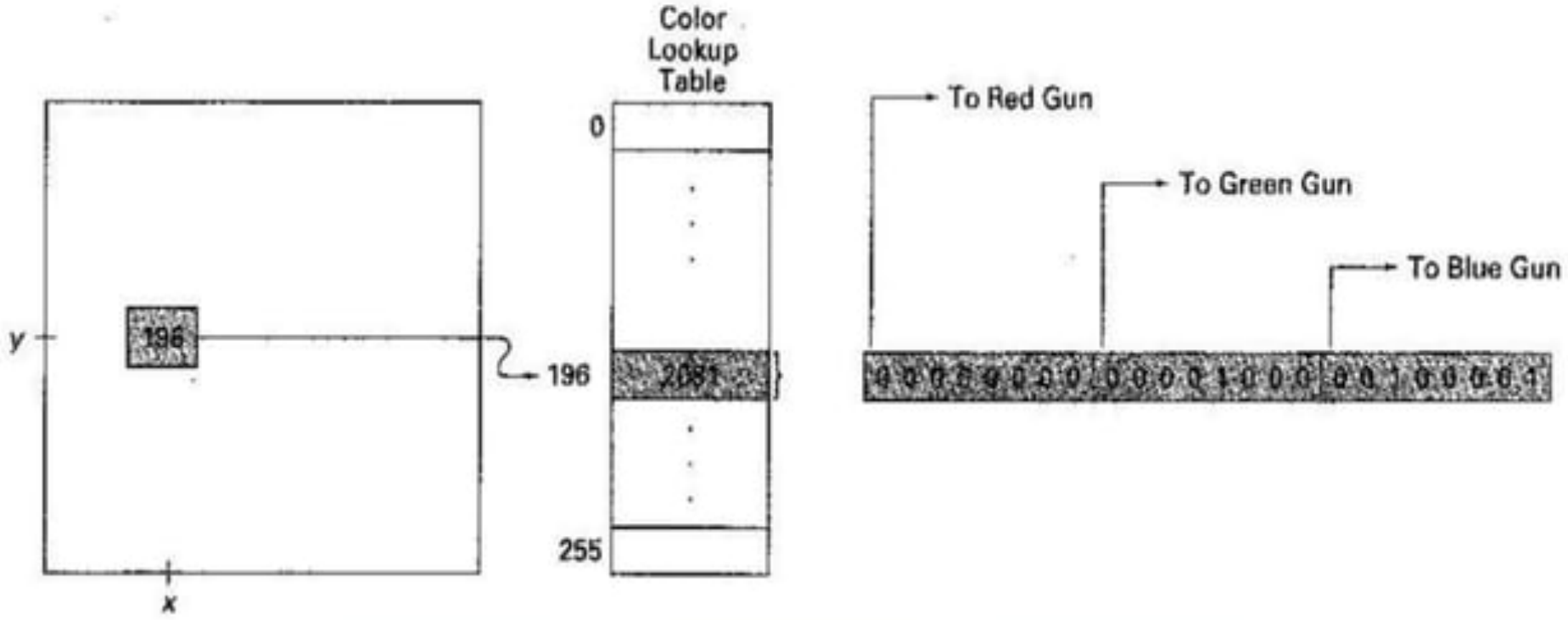
- General Purpose raster scan systems provide a variety of colors while random scan monitors provide very few.
- Colors are represented by colors codes which are positive integers.
- Color information is stored in frame buffer or in separate table and use pixel values as index to the color table.
- Two ways to store color information :
    1. Direct
    2. Indirect

Direct : Color Lookup Table

THE EIGHT COLOR CODES FOR A THREE-BIT PER PIXEL FRAME BUFFER

Color	Stored Color Values in Frame Buffer			Displayed Color
Code	RED	GREEN	BLUE	
0	0	0	0	Black
1	0	0	1	Blue
2	0	1	0	Green
3	0	1	1	Cyan
4	1	0	0	Red
5	1	0	1	Magenta
6	1	1	0	Yellow
7	1	1	1	White

Color Lookup Table



A color lookup table with 24 bits per entry accessed from a frame buffer with 8 bits per pixel. A value of 196 stored at pixel position (x, y) references the location in this table containing the value 2081. Each 8-bit segment of this entry controls the intensity level of one of the three electron guns in an RGB monitor.

GrayScale

Apply for monitor that have no color

Shades of grey (white->light grey->dark grey->black)

Color code mapped onto grayscale codes

2 bits can give 4 level of grayscale

8 bits per pixel will allow 256 combination

Dividing the actual code with 256 will give range of 0 and 1

Ex:
Color code in color display is 118

To map to nearest grayscale then

118/256 = 0.45

➔ light gray

INTENSITY CODES FOR A FOUR-LEVEL GRAYSCALE SYSTEM

Intensity Codes	*Stored Intensity Values In The Frame Buffer (Binary Code)*		*Displayed Grayscale*
0.0	0	(00)	Black
0.33	1	(01)	Dark gray
0.67	2	(10)	Light gray
1.0	3	(11)	White

Area Fill Attributes

Option for filling a defined region is whether solid , pattern and colors.

Fill Styles

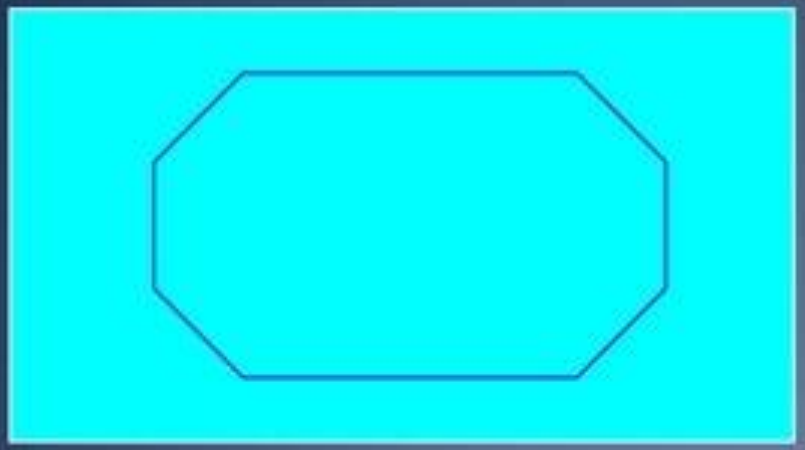
Three basic fill styles are:

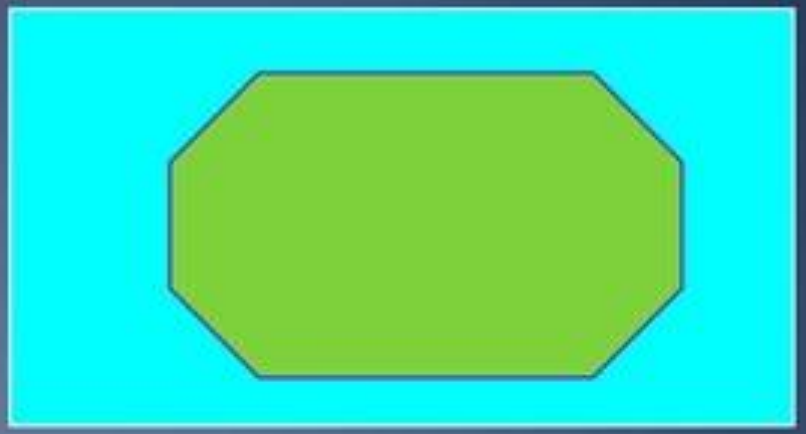
1. Hollow with color border

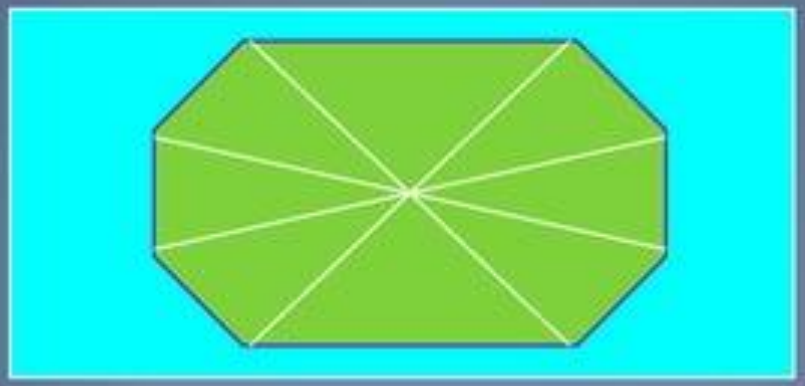
2. Solid

3. Patterened

Area Fill Attributes







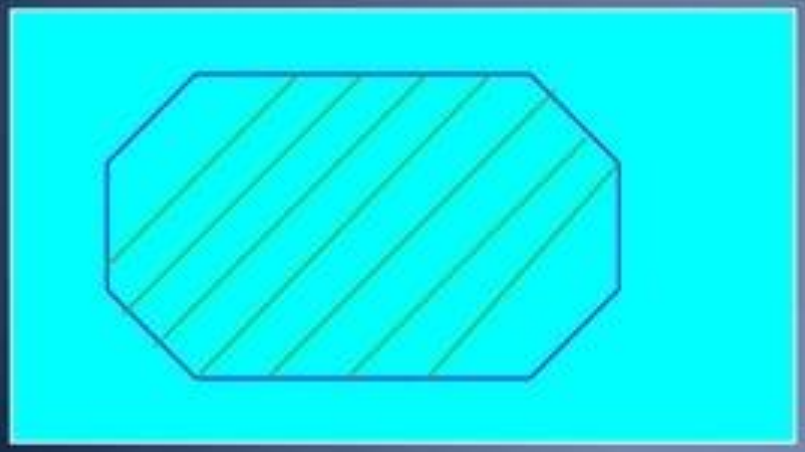
Area Fill Attributes

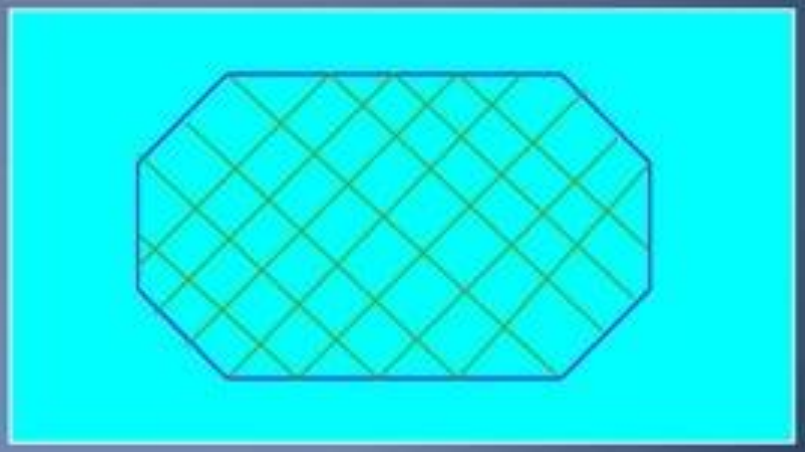
A basic fill style can be assigned in a PHIGS program using the following function:

setInteriorStyle(fs),
Where fs c an be hollow,s olid,or pattern .

Another value for Fill Style is Hatch, which is used to fill an area with selected hatching patterns.2 types :

Diagonal Hatch Fill and Diagonal Cross Hatch Fill.





Area Fill Attributes

The color for a solid interior or for a hollow area outline is chosen with :

setInteriorColorIndex(fc) where fc is the desired color code.

Other fill options include specifications for the edge type, edge width edge color of a region.

Pattern Fill

We select fill patterns with
setInteriorStyleIndex(pi), where pi specifies a table position

For fill style pattern, tables entries can be created on individual output devices with :

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

Where,
pi is pattern index number,
ws is the workstation code,
Cp is the 2d array of color codes with nx columns and ny rows.