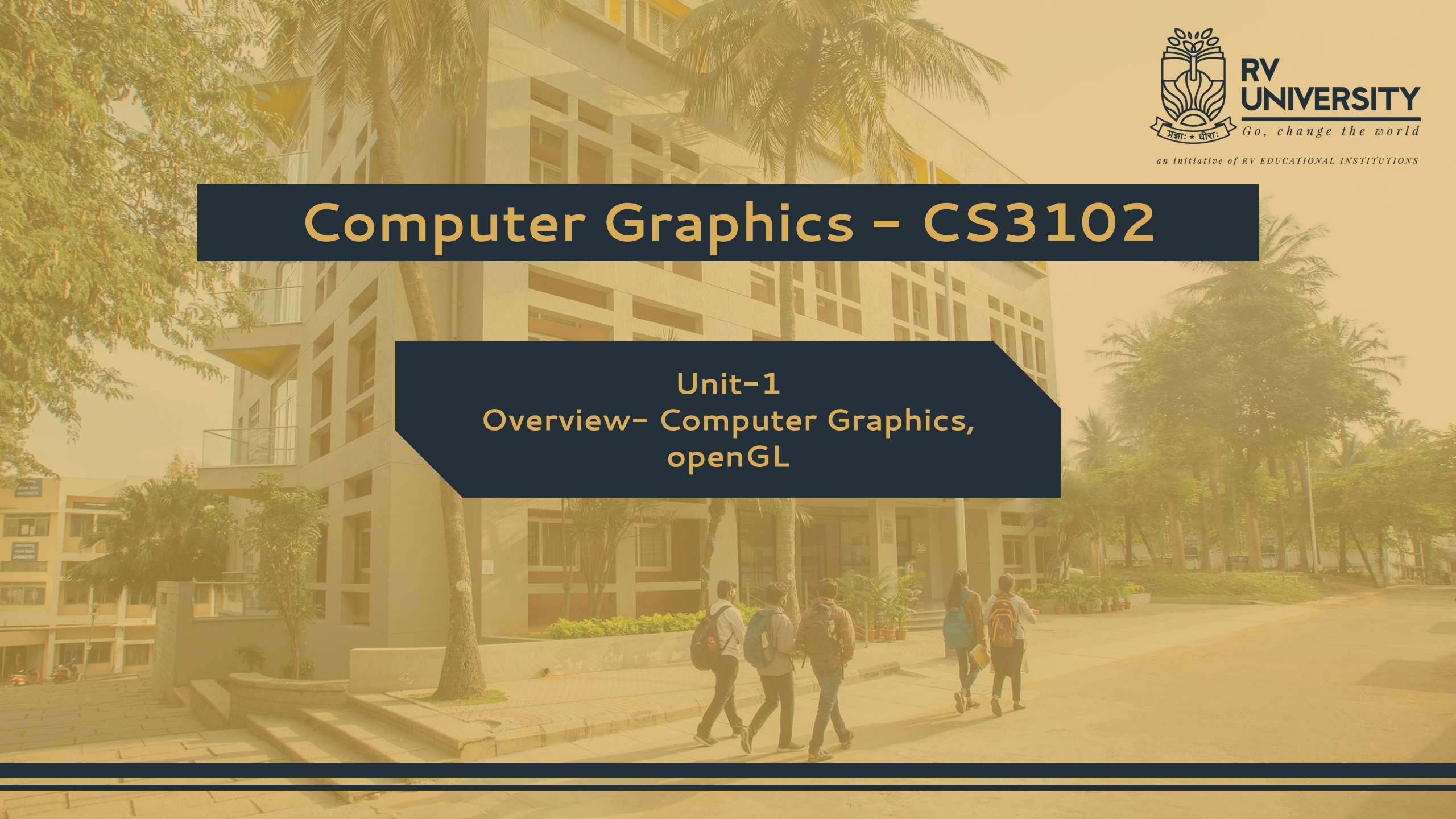


RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Computer Graphics - CS3102

Unit-1
Overview- Computer Graphics, openGL

TEXTBOOKS

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, 3rd / 4th Edition, Pearson Education, 2011
2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Introduction

What is Computer Graphics?

Creation, Manipulation and Storage of geometric objects (modeling) & their images (rendering).

Display those images on screens or hardcopy devices

Applications of Computer Graphics

1. Graphs and charts –Data plotting
2. Computer Aided Design (CAD)
3. Virtual- Reality Environments
4. Data Visualizations
5. Education & Training
6. Computer Art
7. Entertainment (animation, games, …)
8. Image Processing
9. Graphical User Interfaces

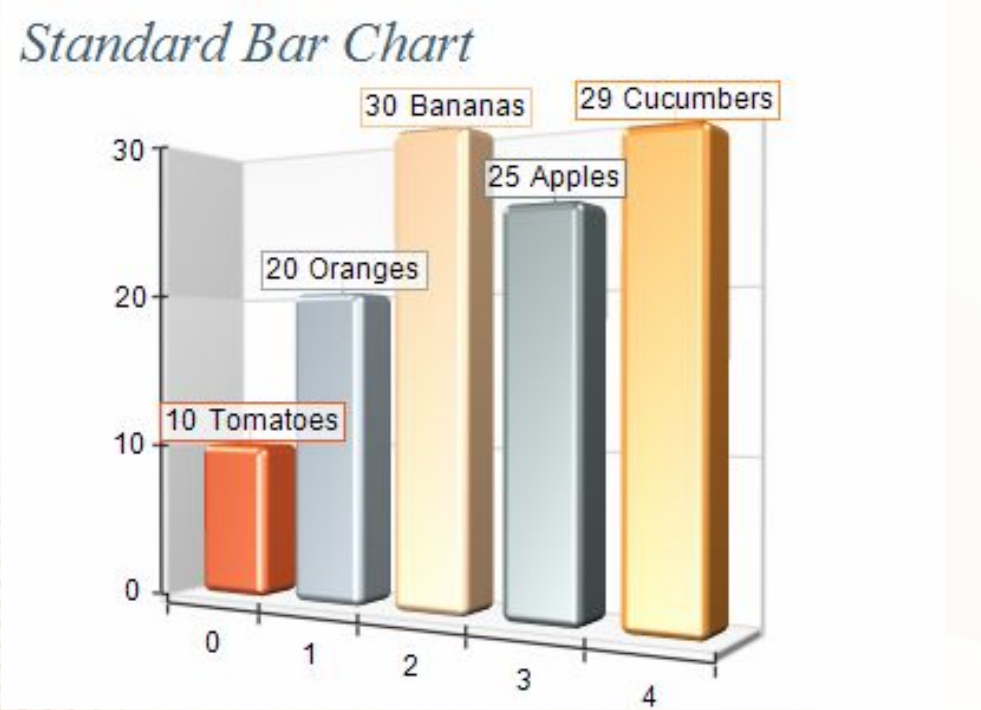
1. Graphs and charts

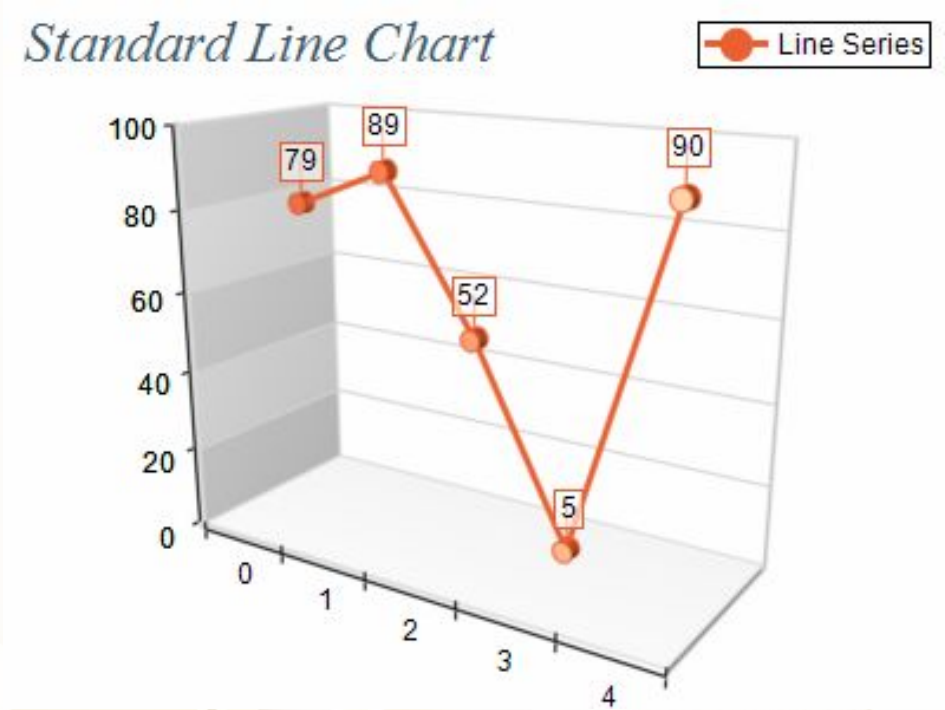
- Display simple data graphs.
- Complex data relationships
- Data plotting – most common application
- Graphs and chart used to summarize:
  - financial ,mathematical ,economic, statistical, engineering, scientific data

Typical eg of data plots

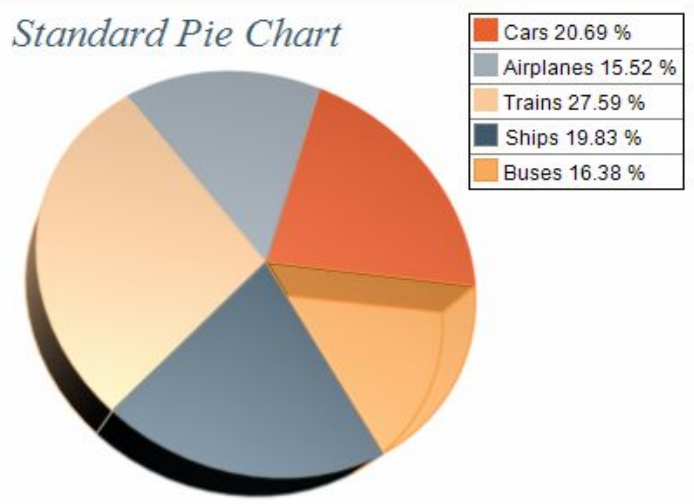
- Line graphs, bar charts, pie charts, surface graph, contour plots or other displays for 2D,3D or higher dimensional

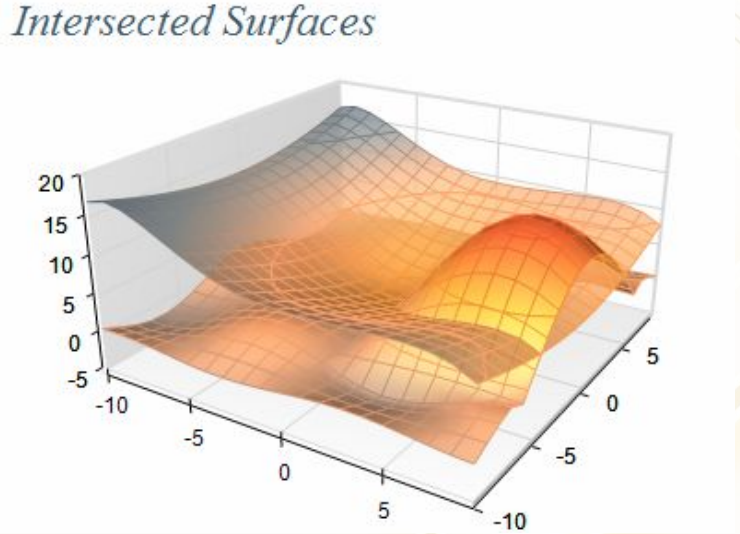
Examples of presentation graphics





Examples of presentation graphics





Time chart used in task planning

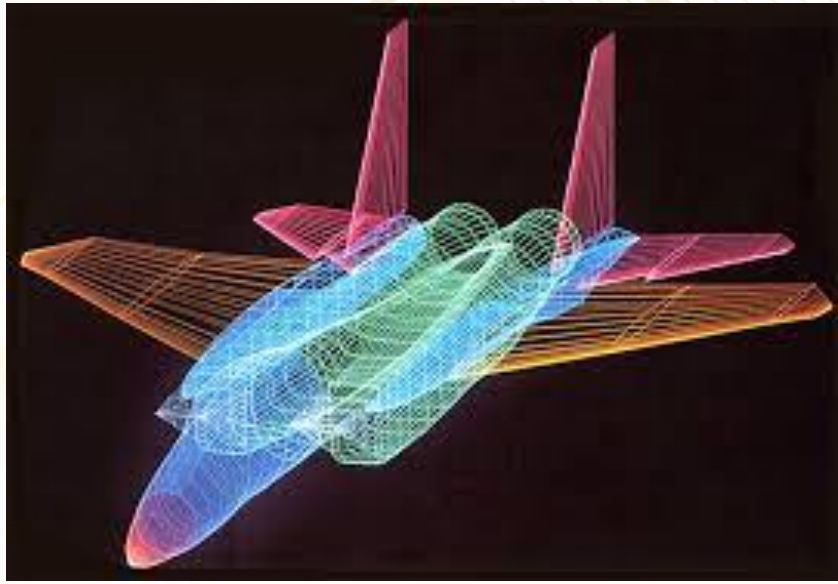
Time chart and task network layouts are used in project management to schedule and monitor the progress of projects.

2.Computer Aided Design (CAD)

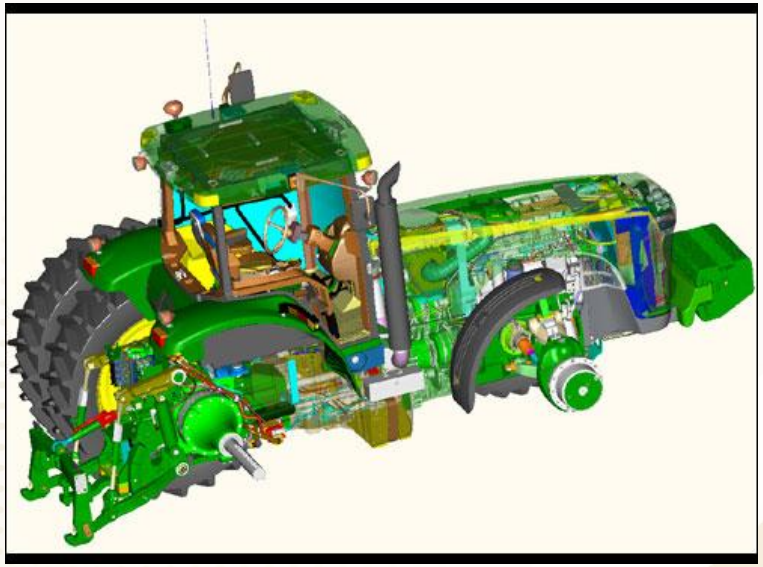
Used in design of buildings, automobiles, aircraft, watercraft, spacecraft, computers, textiles & many other products

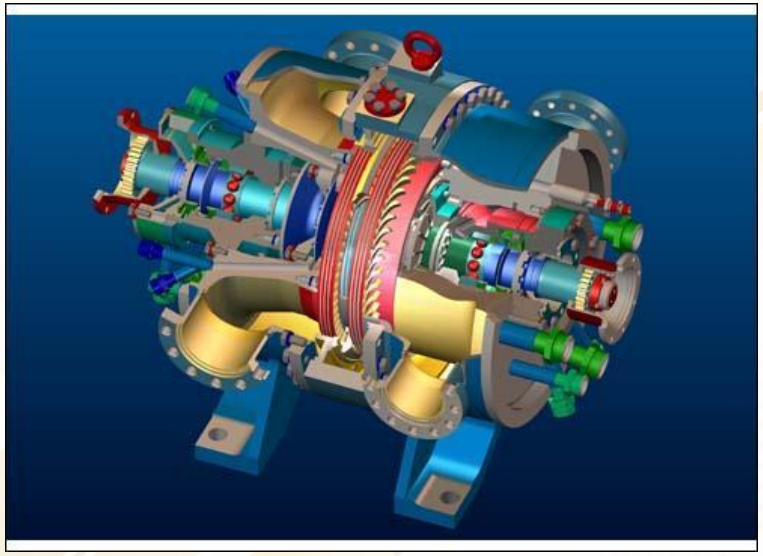
Objects are displayed in **wire frame** outline form-overall shape and internal features

Software packages provide multi-window environment- different views



Graphics design package provides standard shapes (useful for repeated placements)
Animations are also used in CAD applications.
Real time and computer animation using wireframe - quickly test the performance of a vehicle system, interior, inner components.





Circuits and Networks communications
Computer aided manufacturing(CAM)
Architects use CG to lay out floor plans, positioning of rooms, doors, windows, stairs, shelves, counters, wiring, electrical outlets and fire warning systems
Optimize space utilization

3.Virtual- Reality Environment

- A more recent application of CG is in creation of virtual reality environment in which a user can interact with the objects in three dimension scene.
- Specialized H/w devices provide 3D viewing effects and allow the user to pick up objects in a scene.

To Train heavy equipment operators- tractor
To Train pilot, astronaut
Simulated walk through the rooms or around the outsides of buildings.

4.Data Visualization

Scientific Visualization

Producing graphical representations for scientific, engineering, and medical data sets

Large amounts of info is converted to a visual form, the trends and patterns are often immediately apparent.

A collection of data can contain scalar values, vectors, higher-order tensors, or any combination of these data types.

Mathematical curve
A model of ocean floor
Protein modelling
Molecular structure
Air pollution study
A corn growing study
Airflow over the surface of a space shuttle

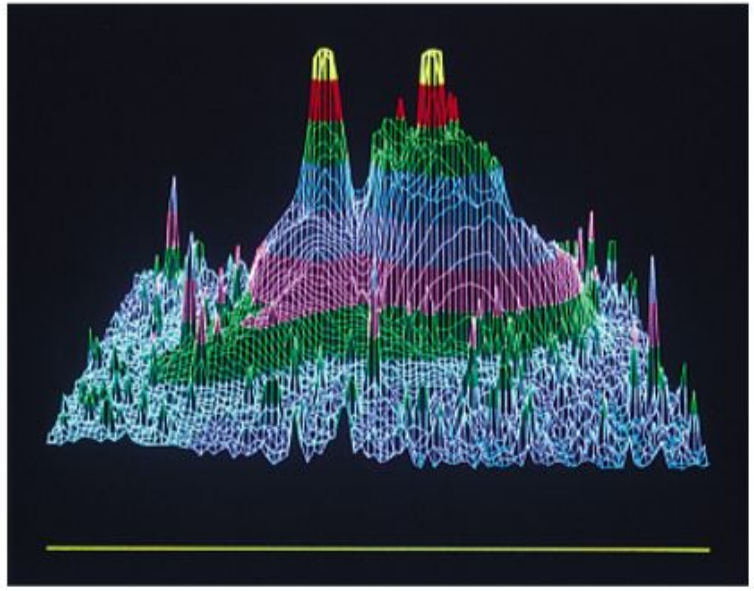


FIGURE 1-23 A color-coded plot with sixteen million density points of relative brightness observed for the Whirlpool Nebula reveals two distinct galaxies. (*Courtesy of Los Alamos National Laboratory.*)

FIGURE 1-24 Mathematical curve functions plotted in various color combinations. (*Courtesy of Melvin L. Prueitt, Los Alamos National Laboratory.*)

Business Visualization is used in connection with data sets related to **commerce, industry and other non-scientific areas**
Techniques used- color coding, contour plots, graphs, charts, surface renderings & visualizations of volume interiors.
Image processing techniques are combined with computer graphics to produce many of the data visualizations.

5.Education & Training

Computer generated models of physical, financial and economic systems are used as educational aids.
Models of physical processes, physiological functions, population trends, or equipment help trainees understand the operation of the system

- Specialized systems used for training applications
  - simulators for practice sessions or training of ship captains
  - aircraft pilots
  - heavy equipment operators
  - air traffic-control personnel

  Screen for visual display of external environment

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Training

6.Computer Art

Used in fine art & commercial art
Includes artist's paintbrush programs, paint packages, CAD packages, desktop publishing s/w and animation packages.
These packages provides facilities for designing object shapes & specifying object motions.
Examples : Cartoon drawing, paintings, product advertisements, logo design

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Examples :

Computer Art

Electronic painting
Picture painted electronically on
a graphics tablet (digitizer) using a **stylus**
Cordless, pressure sensitive stylus
Morphing
A graphics method in which one object is transformed into another

Commercial art

Computer-generated animations are also frequently used in producing television commercials.

Generated frame by frame, where each frame of the motion is rendered and saved as a separate image file.

In each successive frame, object positions are displaced slightly to simulate the motions involved in the animation.

all frames -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, atleast 30 frames per second are required.


Easy Morph
File Objects Generate Display Settings Filmstrip View Help
Animate
Magnify
Generate
One Frame
End - Frogorig.trm
Start - Firegirl.trm
Tools
Result
Filmstrip
Frame
Speed
3 / 5
00:00.080
Ob (unselected At Stop Sequence

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Entertainment

Movie Industry
Used in motion pictures, music, videos, and television shows.
Used in making of cartoon, animation films

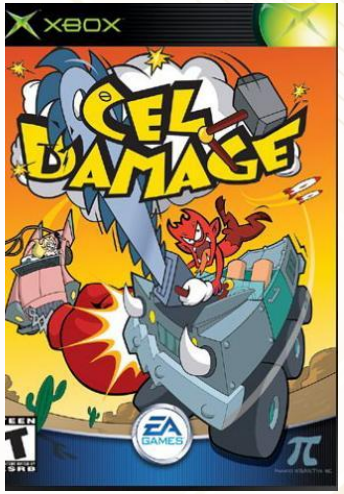
RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Computer Graphics is about animation (films)

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Game Industry

- Focus on interactivity
- Cost effective solutions
- Avoiding computations and other tricks



7. Image Processing

CG- Computer is used to create a picture

Image Processing – applies techniques to modify or interpret existing pictures such as photographs and TV scans.

Improve picture quality, analyze images or recognize visual patterns for robotics app.

Medical applications -Picture enhancements

Tomography

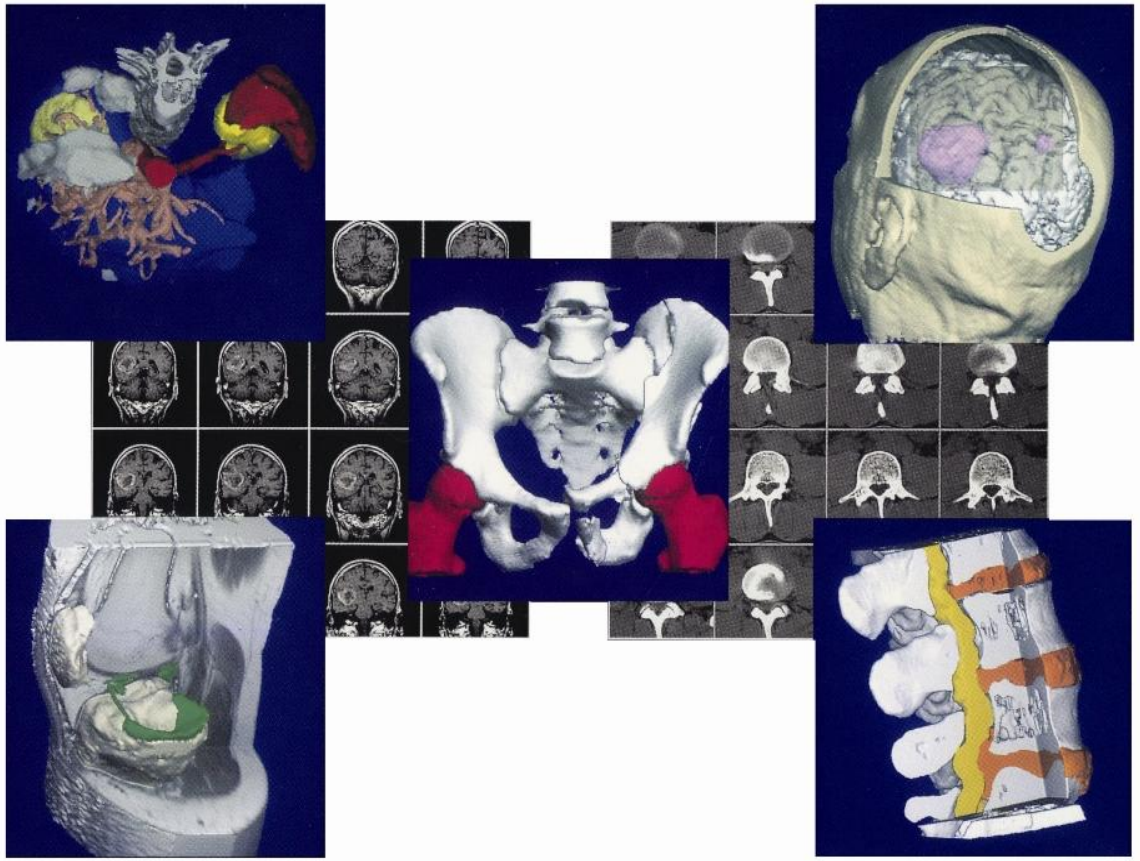
Simulations of surgical operations

Ultrasonics & nuclear medicine scanners

- To apply image processing methods
  - Digitize a photograph (or picture) into an image file
  - Apply digital methods to
    - rearrange picture parts
    - enhance color separations
    - Improve quality of shading
  - Tomography – technique of X-ray photography that allows cross-sectional views of physiological systems to be displayed

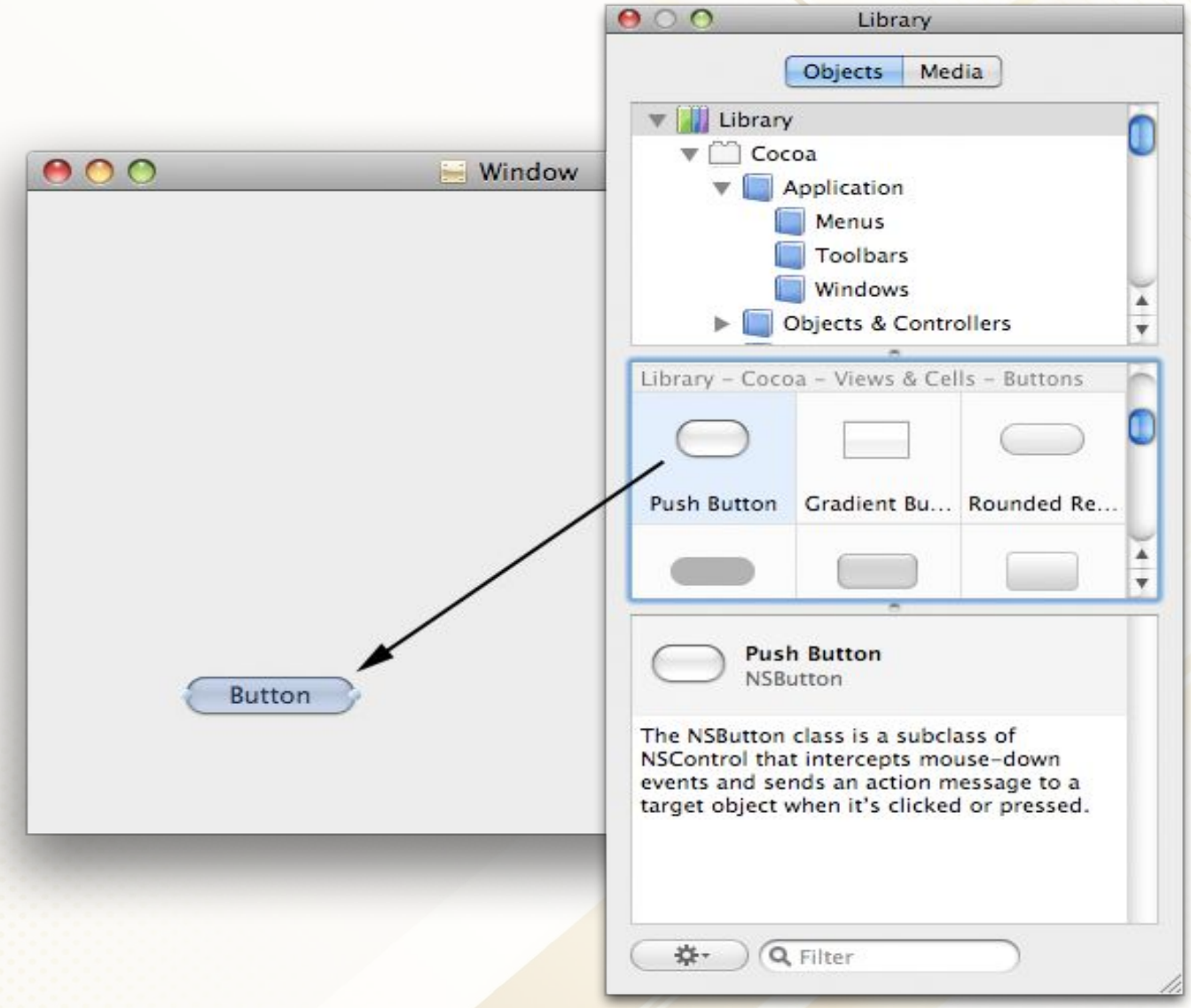
- Computed X-ray tomography (CT) and position emission tomography (PET) use projection methods to reconstruct cross sections from digital data
- **Computer-Aided Surgery** is a medical application technique to model and study physical functions to design **artificial limbs** and to plan & **practice surgery**

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS



8.Graphical User Interfaces

Major component – Window manager (multiple-window areas)
To make a particular window active, click in that window (using an interactive pointing device)
Interfaces display – menus & icons
Icons – graphical symbol designed to look like the processing option it represents
Advantages of icons – less screen space, easily understood
Menus contain lists of textual descriptions & icons

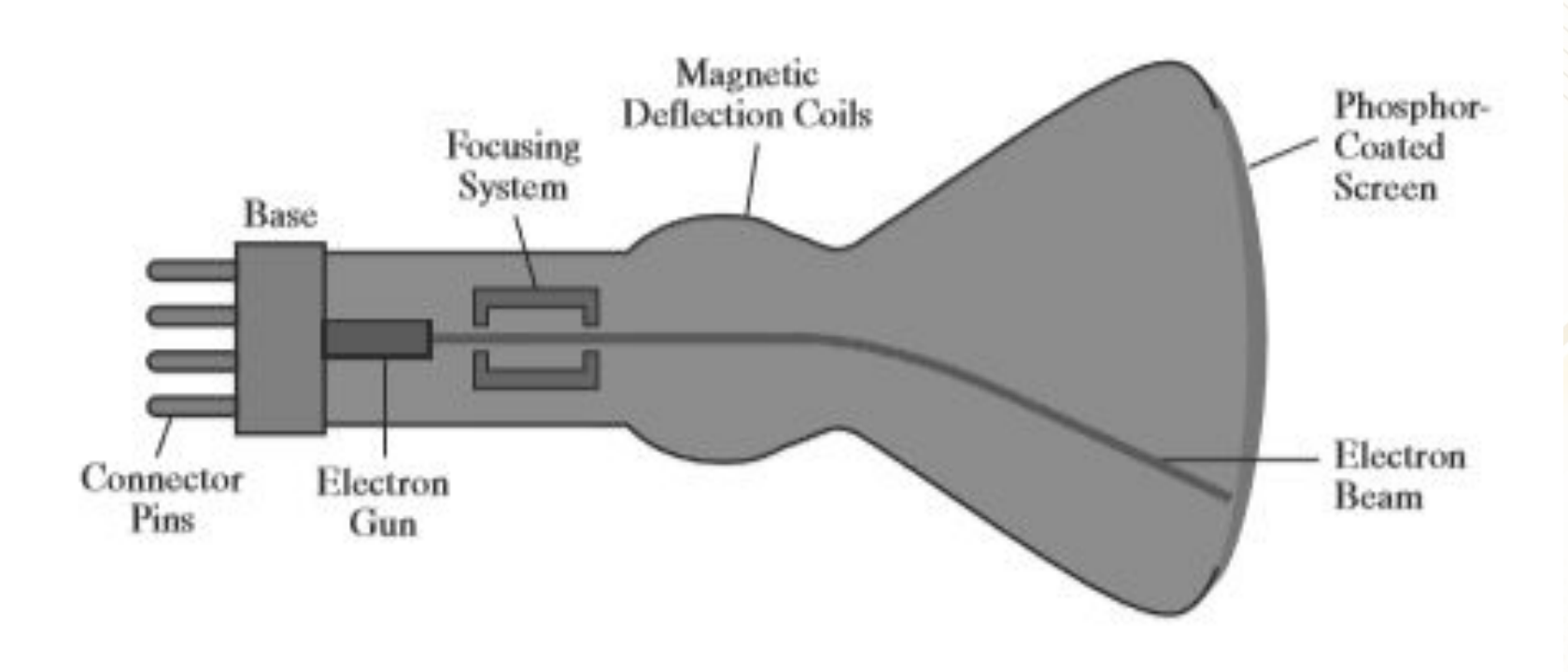

Window
Button
Library
Objects
Media
Library
Cocoa
Application
Menus
Toolbars
Windows
Objects & Controllers
Library – Cocoa – Views & Cells – Buttons
Push Button
Gradient Bu...
Rounded Re...
Push Button
NSButton
The NSButton class is a subclass of NSControl that intercepts mouse-down events and sends an action message to a target object when it's clicked or pressed.
Filter

Video Display Devices

- The primary output device in a graphics system is a **video monitor.**
- The operation of most video monitor is based on the standard Cathode Ray Tube(CRT).
- Solid state monitors -predominate

Refresh Cathode Ray Tubes

Basic design of a magnetic-deflection CRT



- A beam of electrons (cathode rays),emitted by an electron gun, passes through focusing and deflection systems that direct the beam toward specified positions on the phosphor-coated screen.
- The phosphor then emits a small spot of light at each position contacted by the electron beam.
- Because the light emitted by the phosphor fades very rapidly, some method is needed for maintaining the screen picture.

Refresh Cathode Ray Tubes

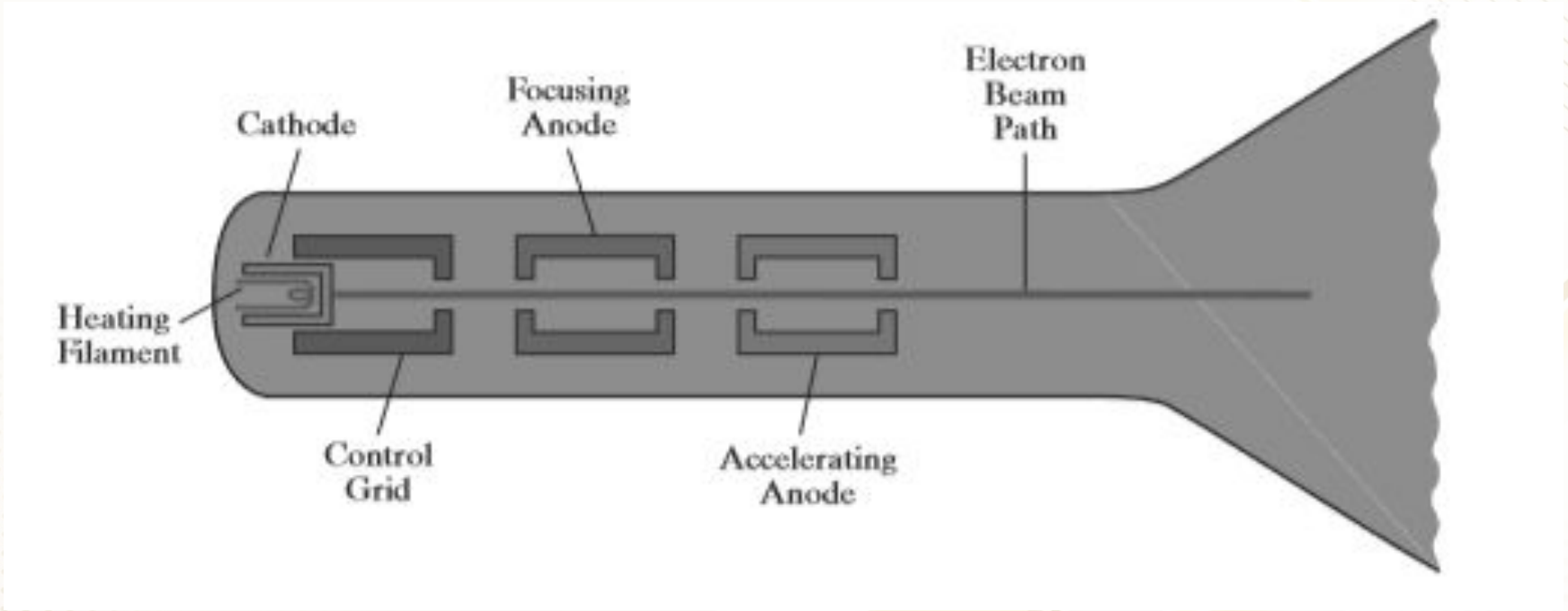
Refresh CRT

- Maintaining phosphor glow is to redraw the picture repeatedly by quickly directing the electron beam back over the same screen points. This type of display is called a refresh CRT

Refresh rate

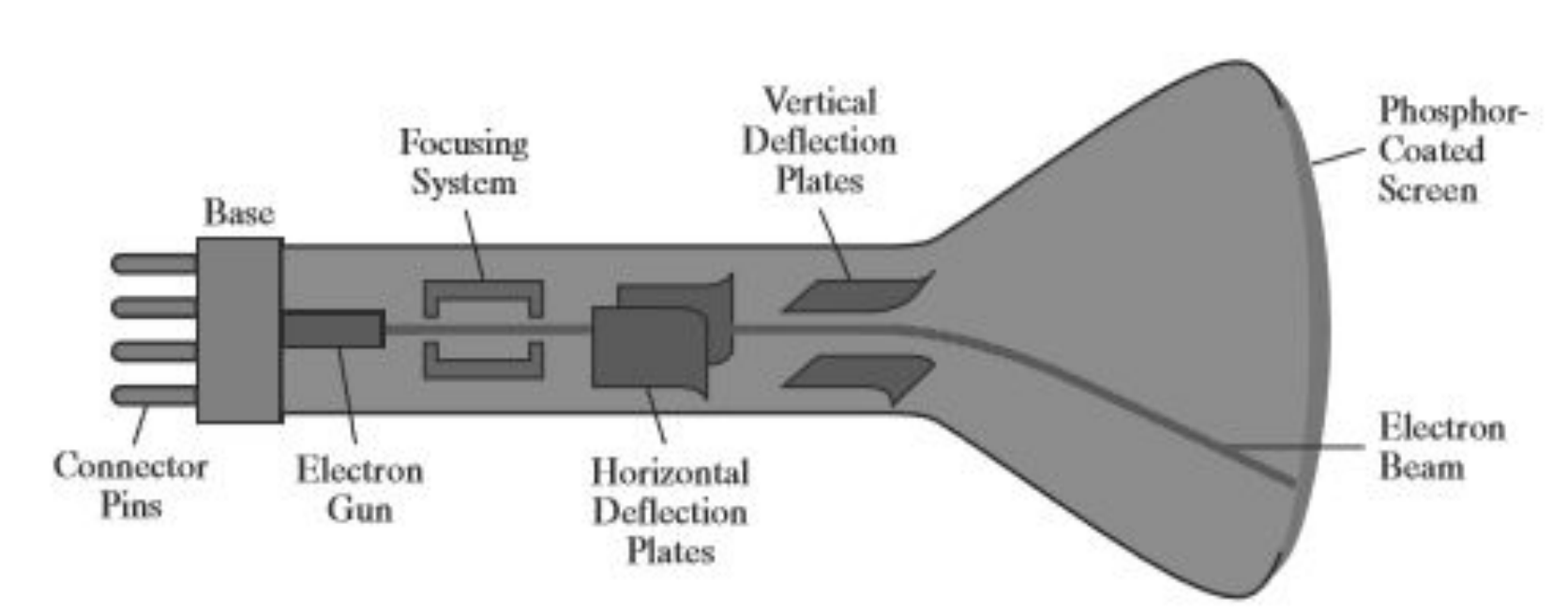
- The frequency at which a picture is redrawn on the screen is referred to as the refresh rate.

Operation of an electron gun with an accelerating anode



RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Electrostatic deflection of the electron beam in a CRT



Refresh Cathode Ray Tubes

- The primary components of an electron gun in a CRT are the heated metal cathode and a control grid
- Heat is supplied to the cathode by directing a current through a coil of wire, called the filament, inside the cylindrical cathode structure.
- Electron are “boiled off” the surface of cathode by heating it with the help of the heating filament.
- Negatively charged electrons are accelerated toward the phosphor coating by a high positive voltage.
- The accelerating voltage is generated with the help of positively charged metal coating on the inside of CRT envelope or an accelerating anode can be used.

Intensity or brightness

Controlled by setting the negative voltage at the control grid.
A high negative voltage - shut off the beam .
A smaller negative voltage on the control grid simply decreases the number of electrons passing through.
Brightness of a display point is controlled by varying the voltage on the control grid.

- Focusing system is needed to force the electron beam to converge into small spot as it strikes the phosphorous.
- Focusing can be implemented with the help of electric field or the magnetic field.
- The electron beam will be focused properly only at the center of the screen.
- As the beam moves to the outer edges of the screen, displayed images become blurred

Deflection of the electron beam can be controlled with either electric or magnetic fields.

CRTs are now commonly constructed with magnetic-deflection coils mounted on the outside of the CRT.

Two pairs of coils are used for this purpose.

One pair is mounted on the top and bottom of the CRT neck, and the other pair is mounted on opposite sides of the neck.

Horizontal deflection is accomplished with one pair of coils, and vertical deflection with the other pair.

The proper deflection amounts are attained by adjusting the current through the coils.
When electrostatic deflection is used, two pairs of parallel plates are mounted inside the CRT envelope.
One pair of plates is mounted horizontally to control vertical deflection.
other pair is mounted vertically to control horizontal deflection

A glowing spot that **quickly fades** after all the excited phosphor electrons have returned to their ground energy level.
The frequency(or color)of the light emitted by the phosphor is in proportion to the energy difference between the excited quantum state and the ground state.

Persistence

How long phosphors continue to emit light after the CRT beam is removed.

defined as the time it takes for the emitted light on the screen to decay one-tenth of its original intensity.

Lower persistence phosphors require high refresh rate.

A phosphors with low persistence -animation

Higher persistence phosphors - highly complex, static pictures.

Graphics monitors persistence value -10 to 60 microseconds

FIGURE 4
Intensity distribution of an illuminated phosphor spot on a CRT screen.

FIGURE 5

The intensity is greatest at the center of the spot, and it decreases with a Gaussian distribution out to the edges of the spot.

Resolution.

- The maximum number of points that can be displayed on the screen without overlap.
- No of points per centimeter that can be plotted horizontally and vertically.

- High resolution system are referred to as high- definition system.
- The physical size of the graphical monitor is given as the length of the screen diagonal.

Aspect ratio

- the ratio of horizontal points to vertical points(vice versa) required to produce equal length lines in both direction on the screen.

Raster Scan Display

- The electron beam is swept across the screen one row at a time top to bottom.
- Each row- **scan line**
- As the electron beam moves across scan line, beam intensity is turned ON and OFF to create the pattern of illuminated spots.

refresh buffer or **frame buffer**.

- Picture definition is stored in the memory area.

- **Frame**
  - Total screen area
- Stored color values are retrieved from refresh buffer as beam moves

pixel or pel(picture element).

- Each screen point

color buffer

- Frame buffer store color values

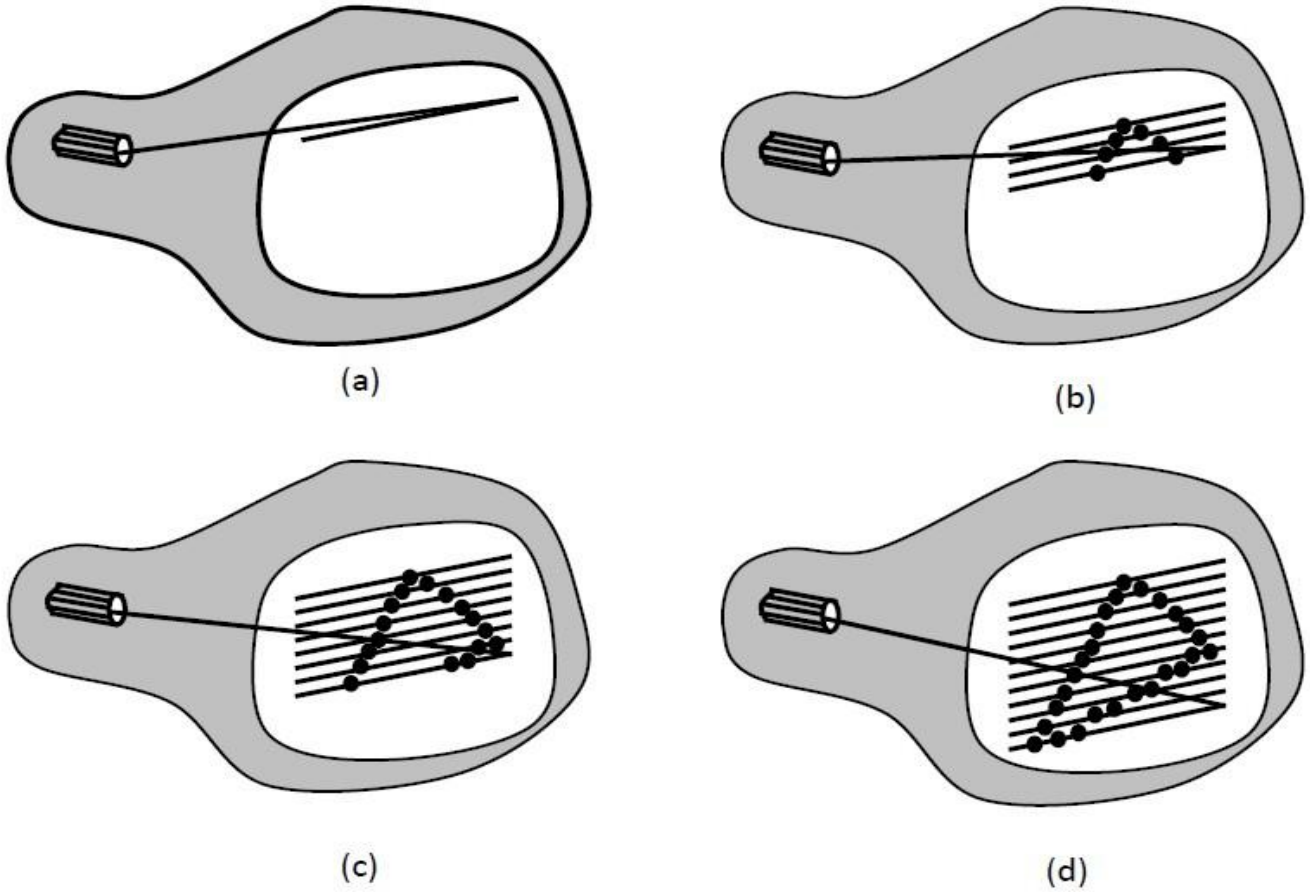


Figure: A raster-scan system displays an object as a set of points across each screen scan line

- Eg of raster systems: Home Television sets, Printers

Aspect ratio:

- No of pixel colums/No of scan lines.

Or

- No of scan lines/No of pixel columns
- No of horizontal points/ No of vertical points
- 4/3 ,Example horizontal line plotted with four points has the same length as a vertical line plotted with three points.

Range of colors- types of phosphor and no of bits per pixel

Bitmap

- On a black white system with one bit per pixel, the frame buffer is commonly called **bitmap**.

Pixmap

- For a system with multiple bits per pixel, the frame buffer is often referred to as **pixmap**.

- 24bits/pixel and resolution 1024by 1024 requires 3megabytes
- Refreshing on current raster scan display **-rate of 60 to 80 frames per second.**
- Below 24 frames/second- flicker

Refresh Rate

- Described in units of cycles per second, or hertz (Hz),
- A cycle corresponds to one frame.
- Refresh rate of 60 frames per second as simply 60 Hz.

Horizontal retrace

- The return to the left of the screen, after refreshing each scan line

Vertical retrace

- And at the end of each frame the electron beam returns to the upper-left corner of the screen to begin the next frame.

Interlaced vs Non-Interlaced Scan

- On some rater scan system and TV , each frame is displayed in two passes using an **interlaced refresh** procedure.
- **Interlaced scan**
  - each frame is displayed in two passes. First pass for odd scan lines and second for the even.

Eg: TV

  - Avoid flickering, provided that adjacent scan lines contain similar display information

- **Non-interlaced scan**
  - electron beam sweep over all the scan lines.

Random scan display (or) vector display (or) stroke- writing (or) calligraphic displays

- Electron beam is directed only to part of the screen where a picture is to be drawn.
- Picture are generated as line drawings
- The component lines of a picture can be drawn and refreshed in any specified order.
- Eg: pen plotter
- **Refresh rate**
  - depends on the number of lines to be displayed.

Refresh display file or display list or display program or simply refresh buffer.

Picture definition is stored as the line drawing commands in memory

Applications

- Suited for line drawing application such as architectural and engg layouts and cannot display realistic shaded scenes.

Advantages

- Vector display has higher resolution than raster scan display.
- produce smooth line drawing as beam directly follow the line path.

Raster systems produce jagged lines

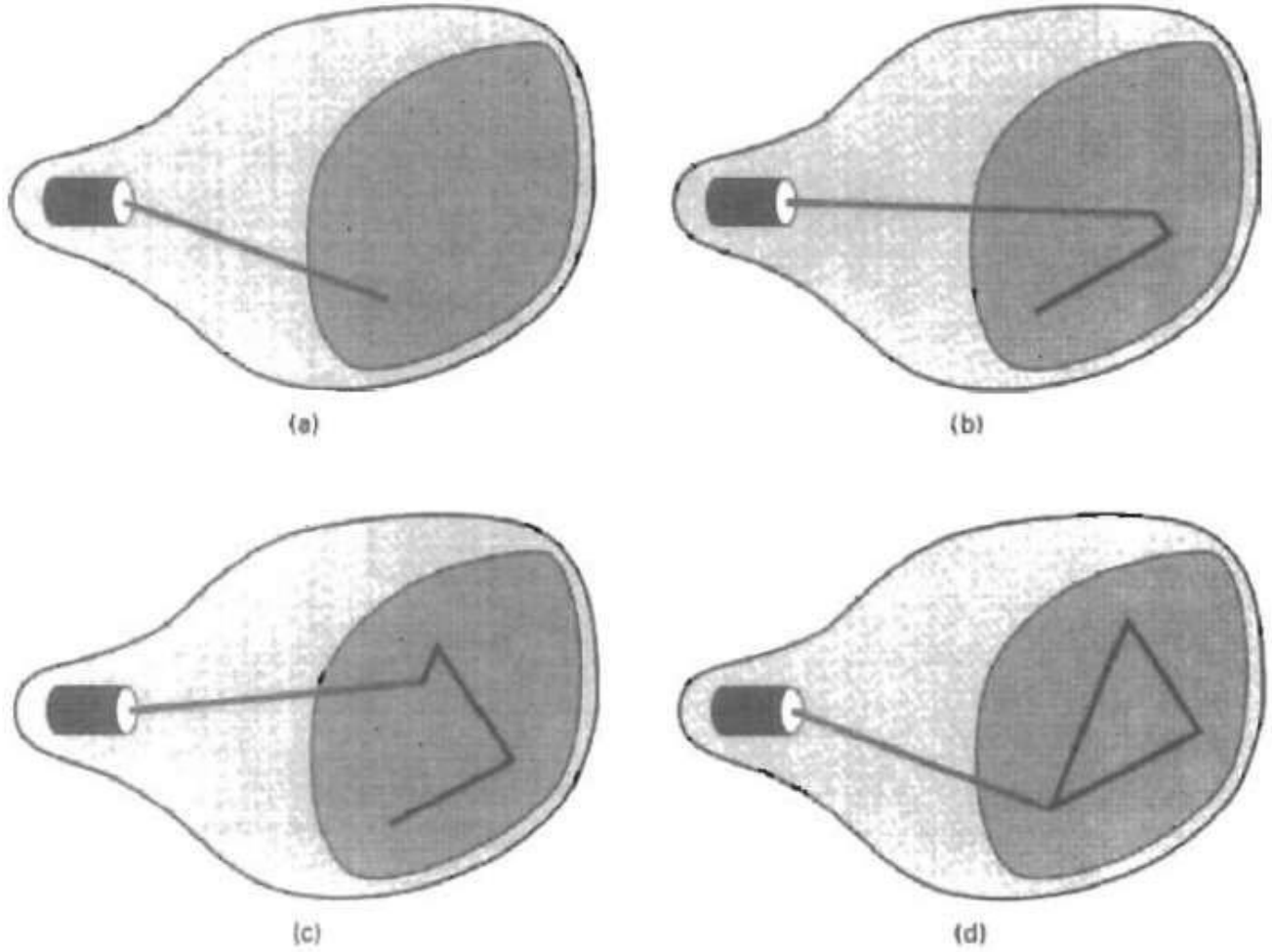


Figure
A random-scan system draws the component lines of an object in any order specified.

Color CRT Monitors

- Color pictures can be displayed by using a combination of phosphors that emit different colored light.
- By combining these different light, range of colors can be generated.
- Two basic techniques used are:
  - **Beam Penetration**
  - **Shadow Mask**

Beam Penetration

- Two layers of phosphors, usually red and green are coated inside CRT.
- Emitted color depends on how far the electron beam penetrates into the phosphor layers.
  - Beam of slow electrons excite only outer red layer.
  - fast beam penetrates through red layer and excites inner green layer.
  - At intermediate beam speeds, orange and yellow.
  - The speed of the electrons, and hence the screen color at any point, is controlled by the beam acceleration voltage.

- inexpensive way to produce color,
- Only a limited number of colors are possible.
- Picture quality is not as good as with other methods.

Shadow Mask

- used in raster scan system(including color TV).
- wider range of colors.
- Three phosphor color dots(red, green, blue) at each pixel position.
- Three electron gun one for each color dot and shadow mask grid just behind the phosphors coated screen.
- The three electron beams are deflected and focused as group onto shadow mask, which contain series of holes aligned with phosphors dot pattern.

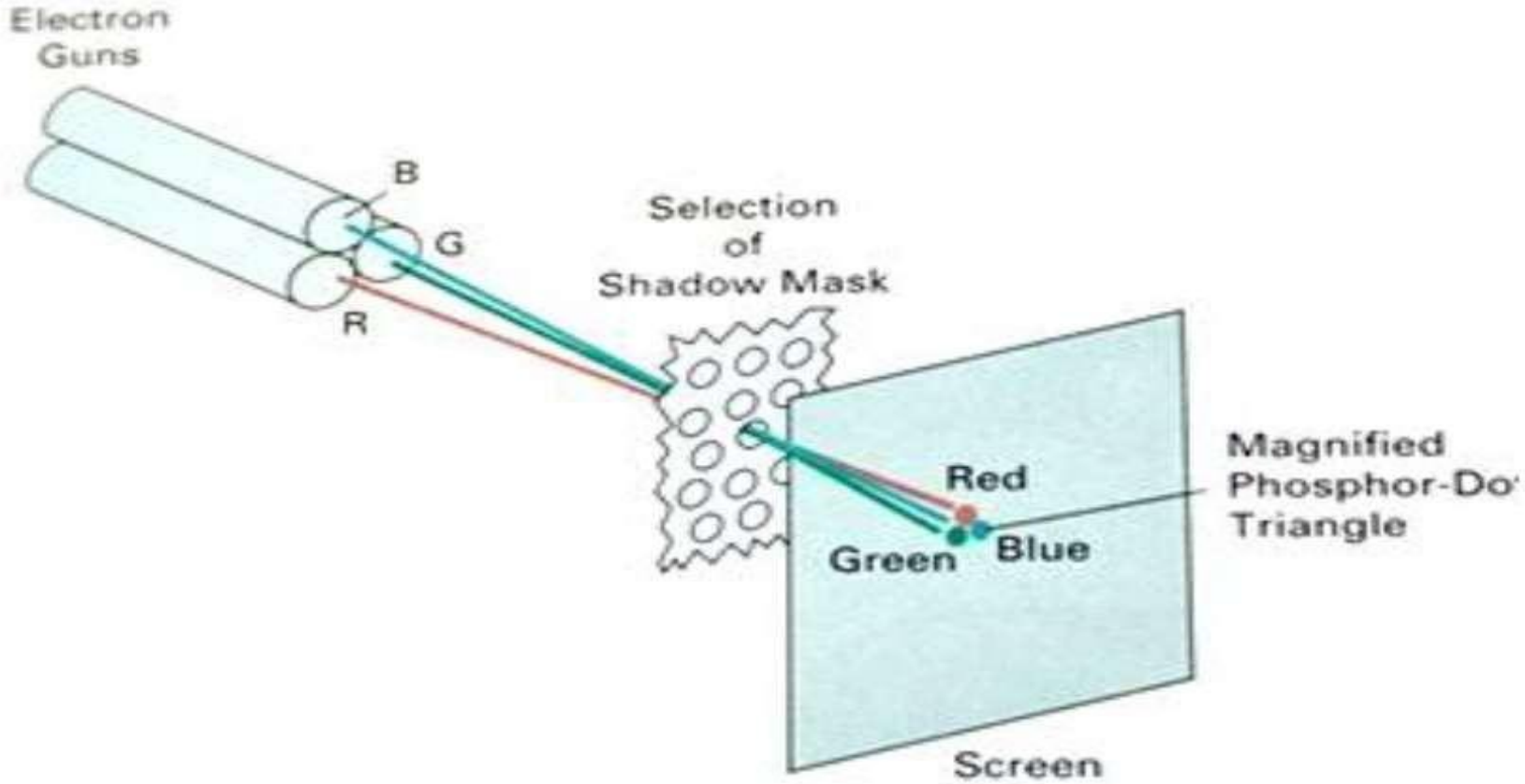


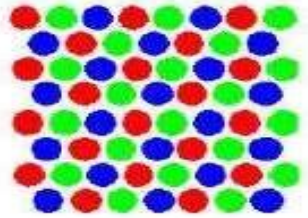
Figure: Shadow mask in Delta-Delta CRT

There are two primary variation:

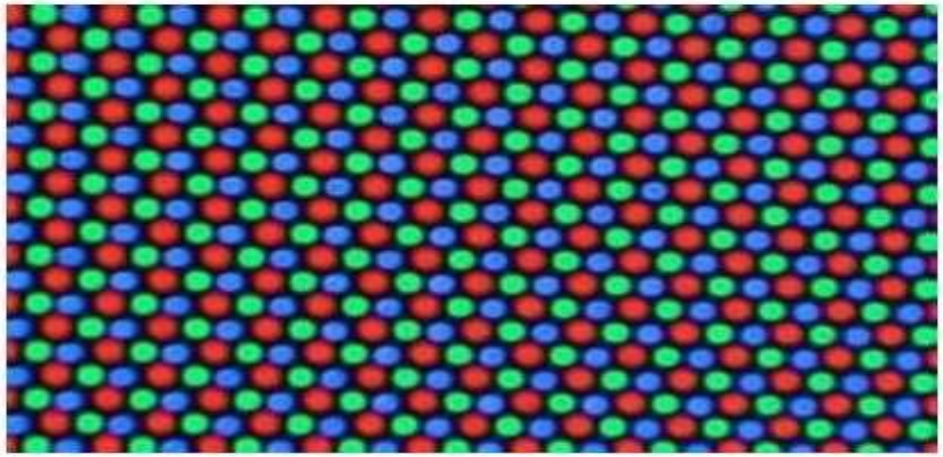
Delta Pattern

In-line

Delta Pattern

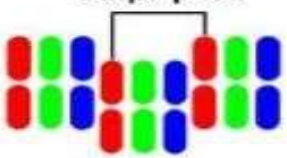


Monitor view



Stripe Pattern

Stripe pitch

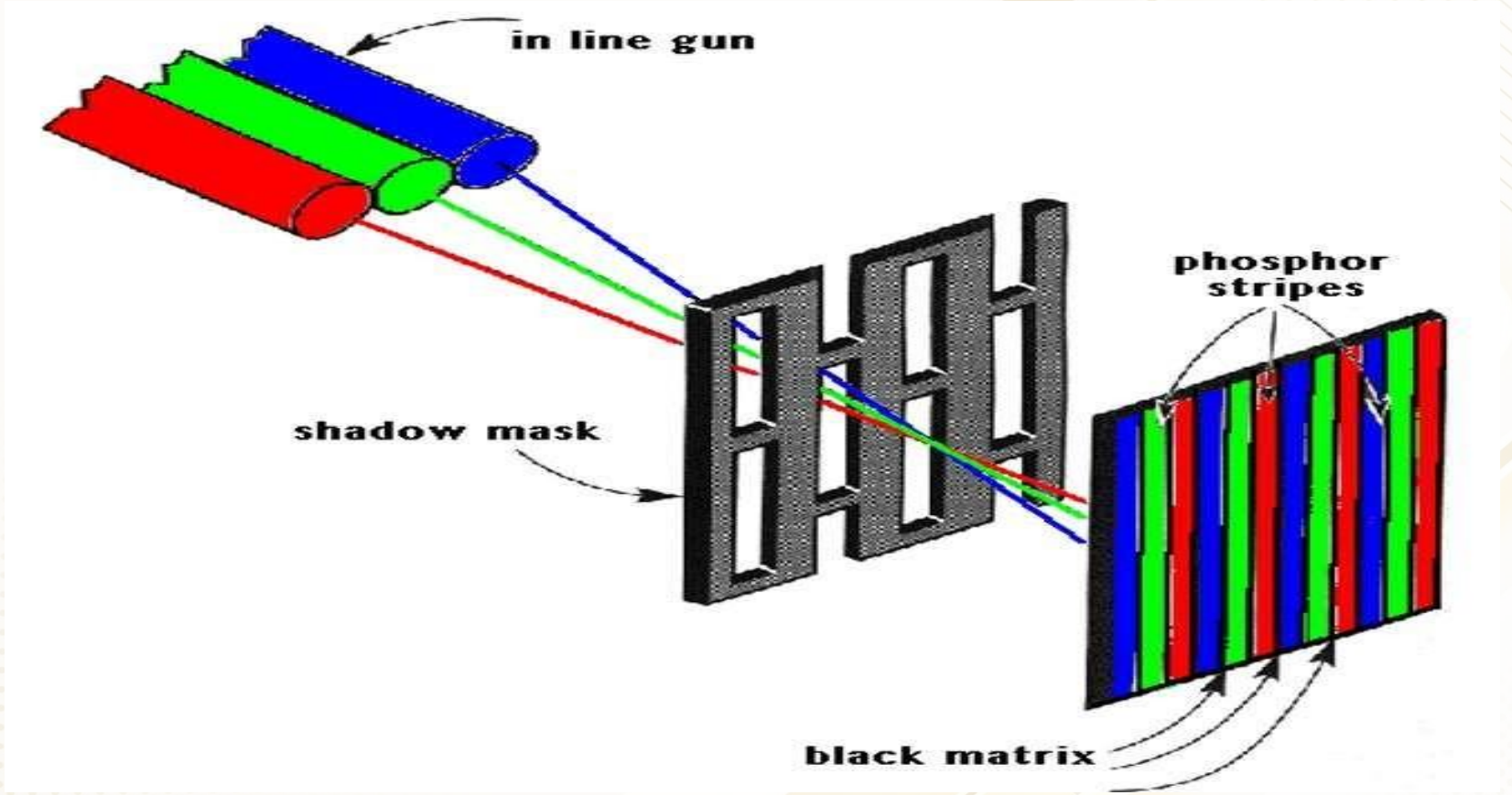


Monitor view

Delta-delta shadow mask CRT

- When three beam pass through the hole in the shadow mask, they activate the dot triangle, which appear as a small color dot on screen.
- Activate only its corresponding color.
- Various colors can be generated by varying the intensity of the three electron guns.
- Difficulties are faced while aligning the shadow mask hole and respective triads.

Inline CRT



- Three electron guns, and RGB color dots on the screen, are aligned along one scanline instead of in a triangular pattern.
- Easier to keep in alignment.
- Commonly used in high-resolution color CRTs.
- Color CRTs are designed with RGB monitors.

Full-color system or a true color system

- An RGB color system with 24bits of storage per pixel
- 17 million color choices.

Flat Panel display

- Class of video devices that have reduced volume, weight and power requirements compared to CRT.
- Thinner, hang them on walls.
- They are used in calculators, pocket video games, laptops, Tv monitors etc.
- We can separate flat panel displays into two categories:
  - **Emissive displays**
  - **Non-emissive displays**

- **Emissive displays(Emitters)**
  - devices that convert the electrical energy into light. Examples are plasma panel, thin-film electroluminescent displays, LED etc.
- **Non-emissive displays(Nonemitters)**
  - optical effects to convert sunlight or light from some other source into graphical patterns. Example LCD

Plasma Panels (or)Gas-discharge displays

- Constructed by filling the region between two glass plates with a mixture of gases that usually includes neon.
- A series of vertical conducting ribbons is placed on one glass panel, and a set of horizontal ribbons is built into the other glass panel.
- Firing voltages applied to a pair of horizontal and vertical conductors cause the gas at the intersection of the two conductors to break down into a glowing plasma of electrons and ions.

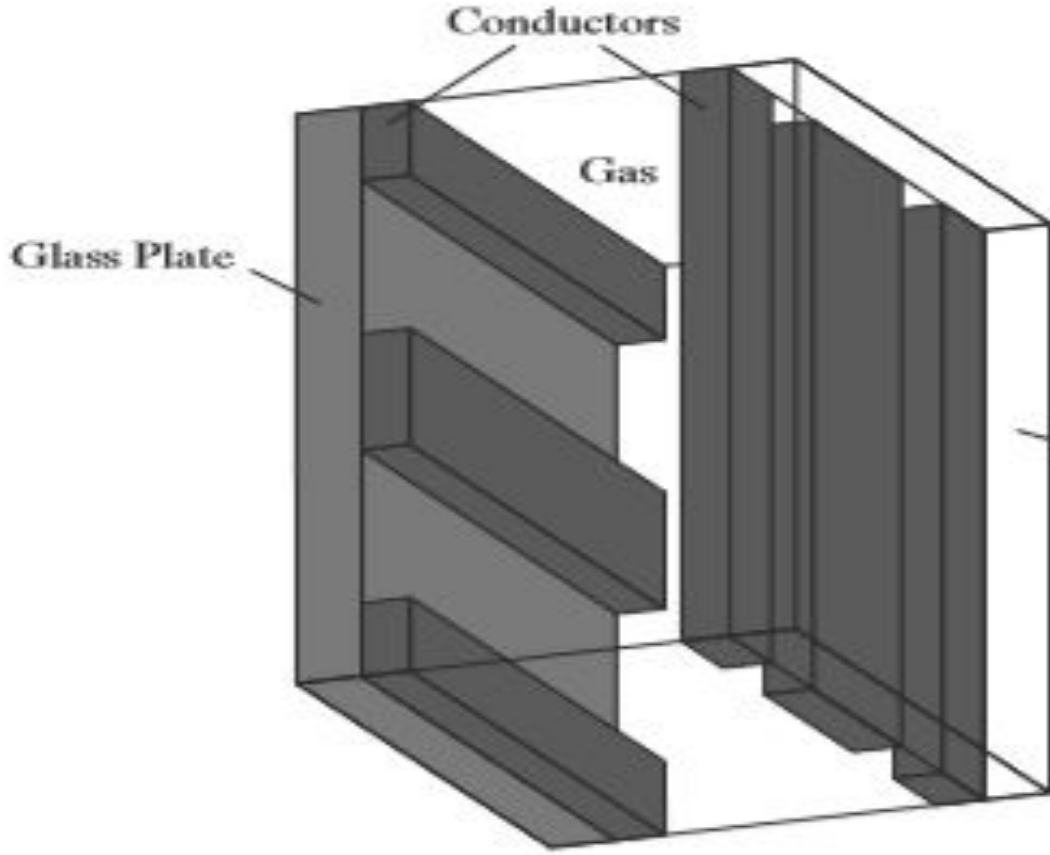


FIGURE 10
Basic design of a plasma-panel display device.

Plasma Panels (or)Gas-discharge displays

- Picture definition is stored in a refresh buffer.
- firing voltages are applied to refresh the pixel positions (at the intersections of the conductors) 60 times per second.

Disadvantage

- Earlier- strictly monochromatic devices.
- Multi color

Thin-film electroluminescent displays
Region between the glass plates is filled with a phosphor, such as zinc sulfide doped with manganese, instead of a gas.
phosphor becomes a conductor in the area of the intersection of the two electrodes.
Electrical energy is then absorbed by the manganese atoms, which then release the energy as a spot of light.
Electroluminescent displays require more power than plasma panels, and good color displays are hard to achieve.

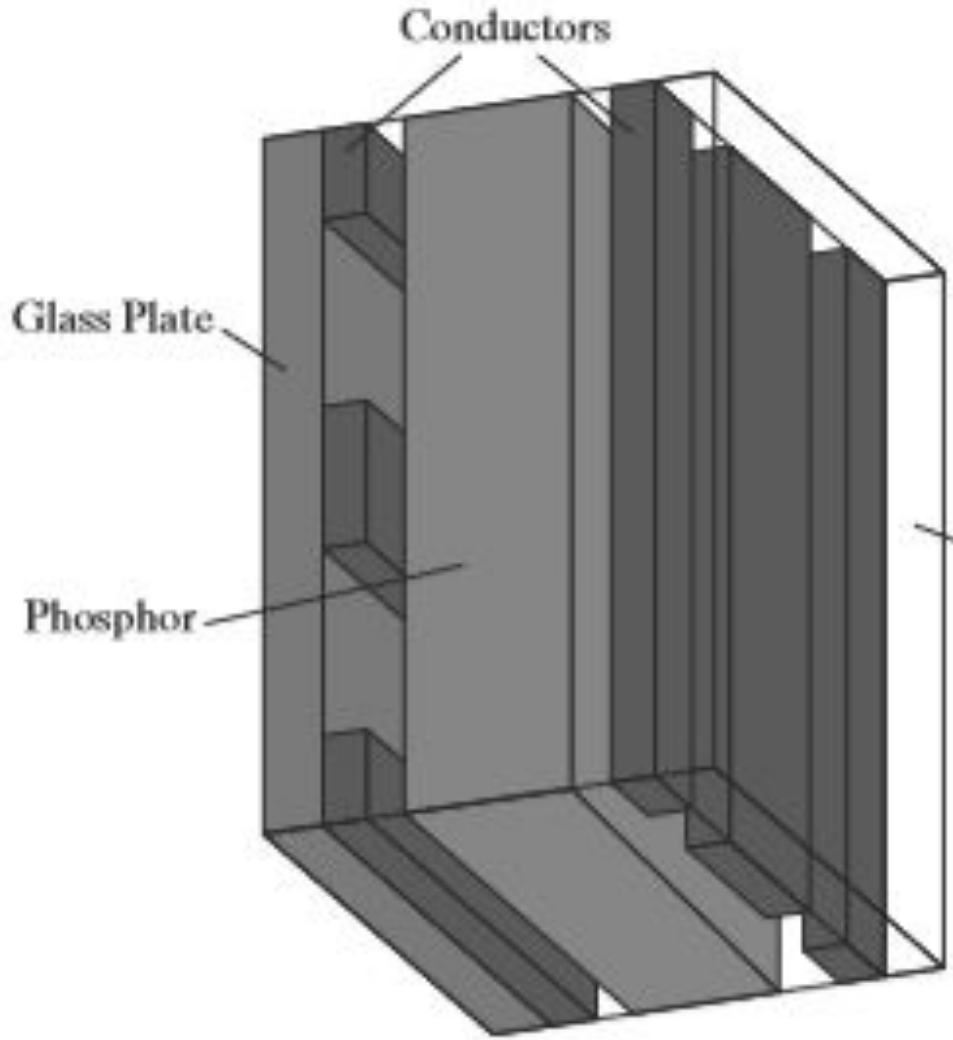


FIGURE 11
Basic design of a thin-film electroluminescent display device.

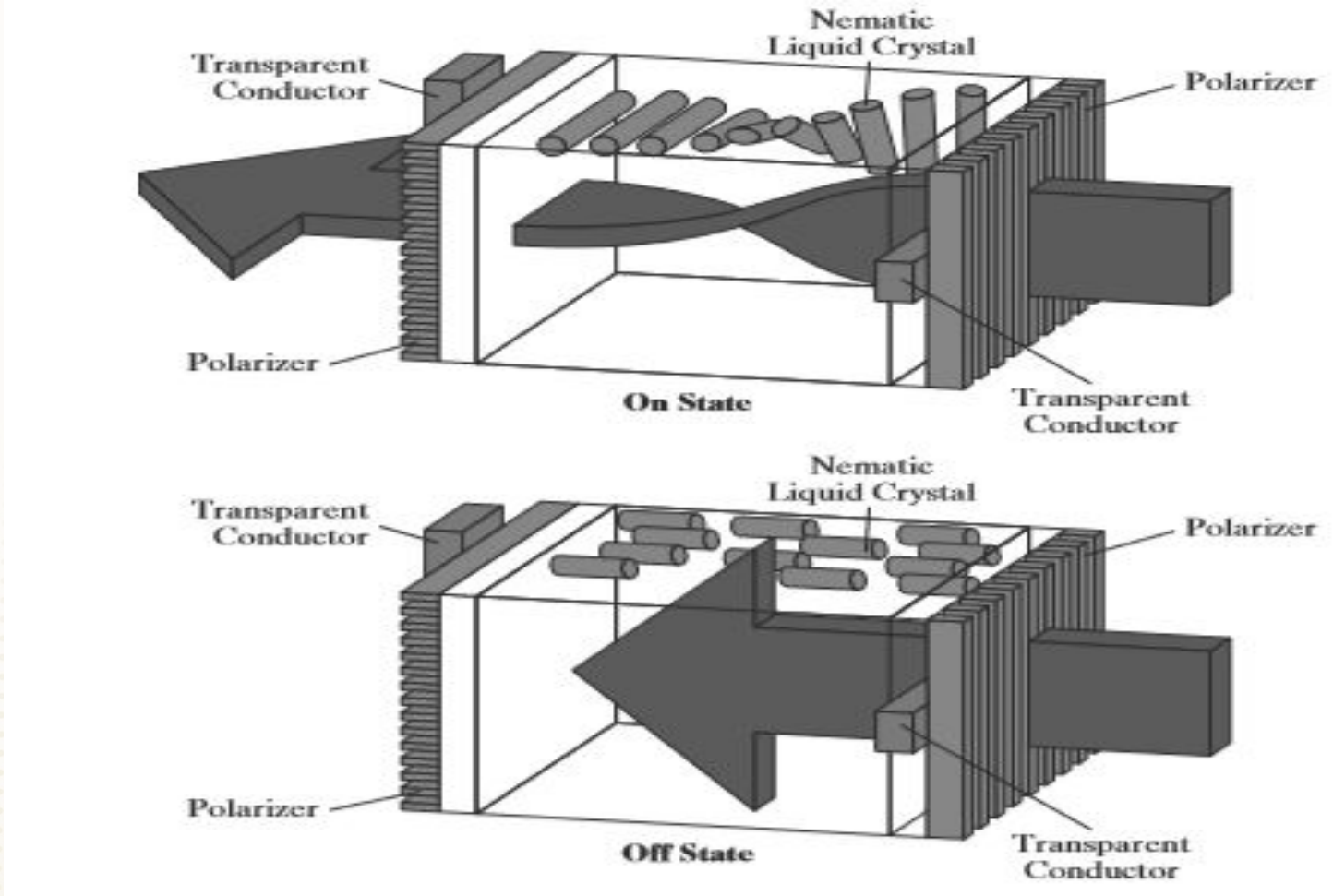
LED (light emitting diode)

A matrix of diodes is arranged to form the pixel.

Information is read from the refresh buffer and converted to voltage levels that are applied to the diodes to produce the light patterns in the display

LCD (liquid crystal displays)

Used in small systems, such as calculators and portable, laptop computers.
non emissive devices
produce a picture by passing polarized light from the surroundings or from an internal light source through a liquid-crystal material that can be aligned to either block or transmit the light.
liquid crystal refers -**compounds have a crystalline arrangement of molecules, yet they flow like a liquid**


Nematic
Liquid Crystal
Transparent
Conductor
Polarizer
Polarizer
On State
Transparent
Conductor
Nematic
Liquid Crystal
Transparent
Conductor
Polarizer
Polarizer
Off State
Transparent
Conductor

Flat-panel displays commonly use nematic(threadlike) liquid-crystal compounds that tend to keep the long axes of the rod-shaped molecules aligned.

A flat-panel display can then be constructed with a nematicliquid crystal

Two glass plates, each containing a light polarizer at right angles to the-other plate, sandwich the liquid-crystal material.

Rows of horizontal transparent conductors are built into one glass plate, and columns of vertical conductors are put into the other plate.

The intersection of two conductors defines a pixel position.

passive-matrix LCD.

Normally, the molecules are aligned as shown in the "on state"
Polarized light passing through the material is twisted so that it will pass through the opposite polarizer.
The light is then reflected back to the viewer.
To turn off the pixel, we apply a voltage to the two intersecting conductors to align the molecules so that the light is not twisted.

Refreshed at the rate of 60 frames per second.

Active matrix LCD

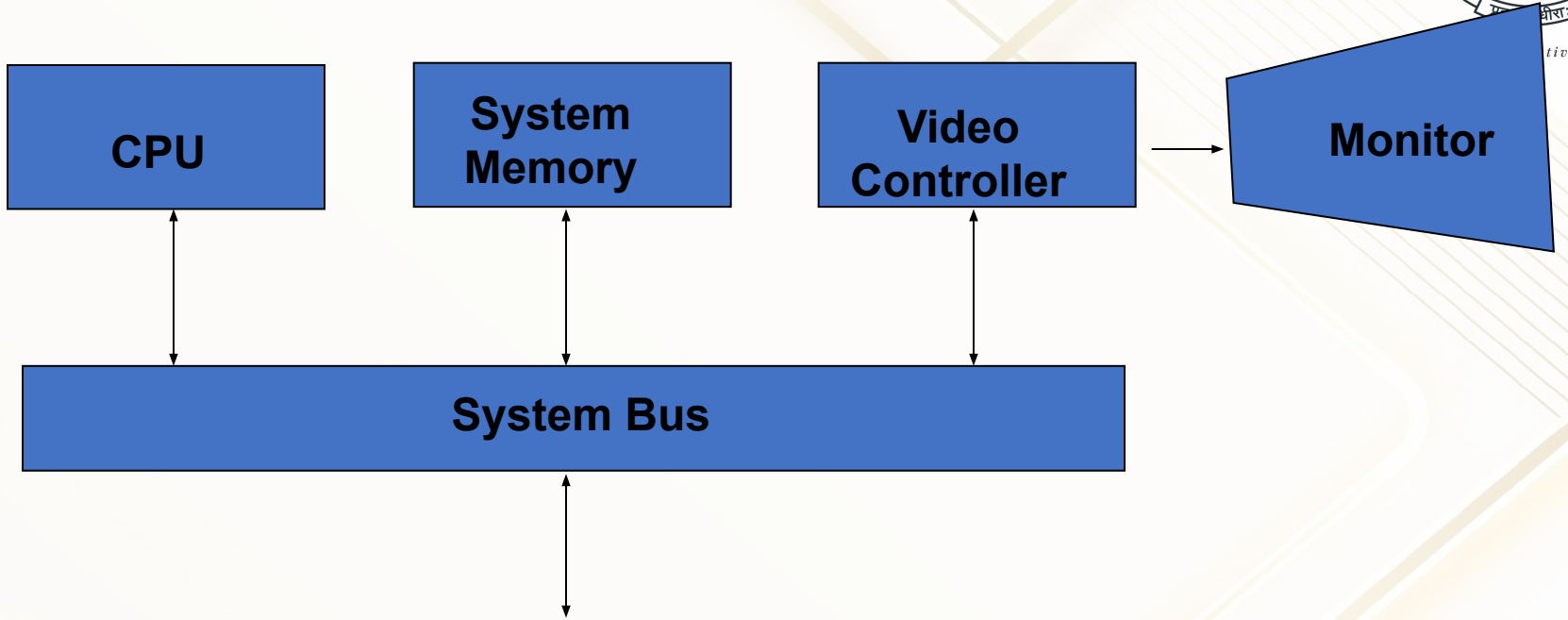
Place a transistor at each pixel location, using thin-film transistor.
Transistor is used to control the voltage at pixel locations.

Raster-scan system

video controller or display controller

- A special purpose processor which is used to control the operation of the display device.
- In addition to video controller, Coprocessors and accelerators.

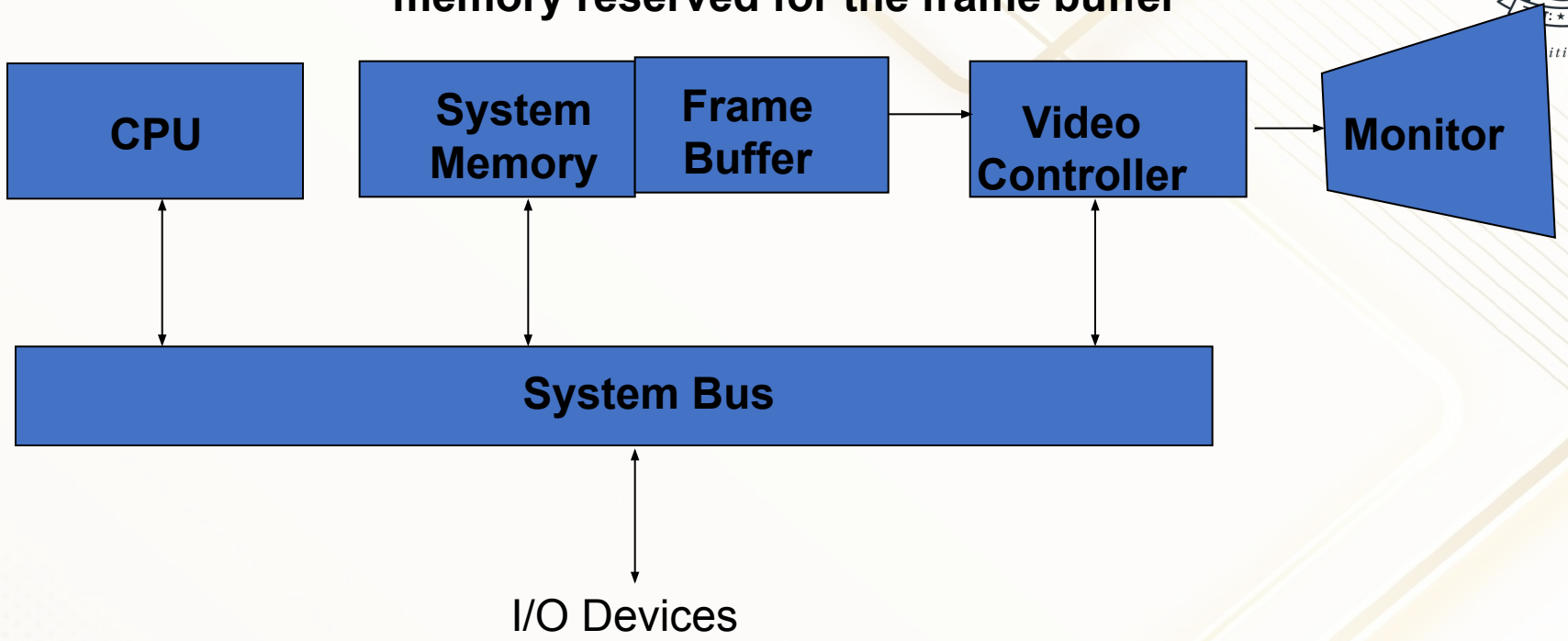
Raster-scan system



Architecture of Simple Raster graphics system

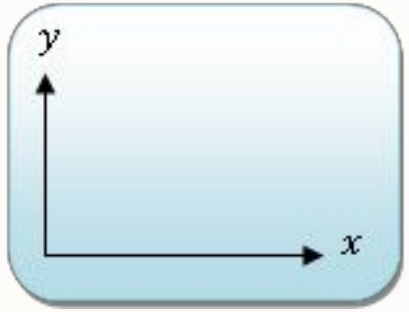
The frame buffer can be any where in the system memory, and the video controller accesses the frame buffer to refresh the screen.

Architecture of Raster system with a fixed portion of the system memory reserved for the frame buffer



- A fixed area of the system memory is reserved for the frame buffer.
- video controller is given direct access to the frame-buffer memory.

Frame-buffer locations, and the corresponding screen positions, are referenced in **Cartesian coordinates**.



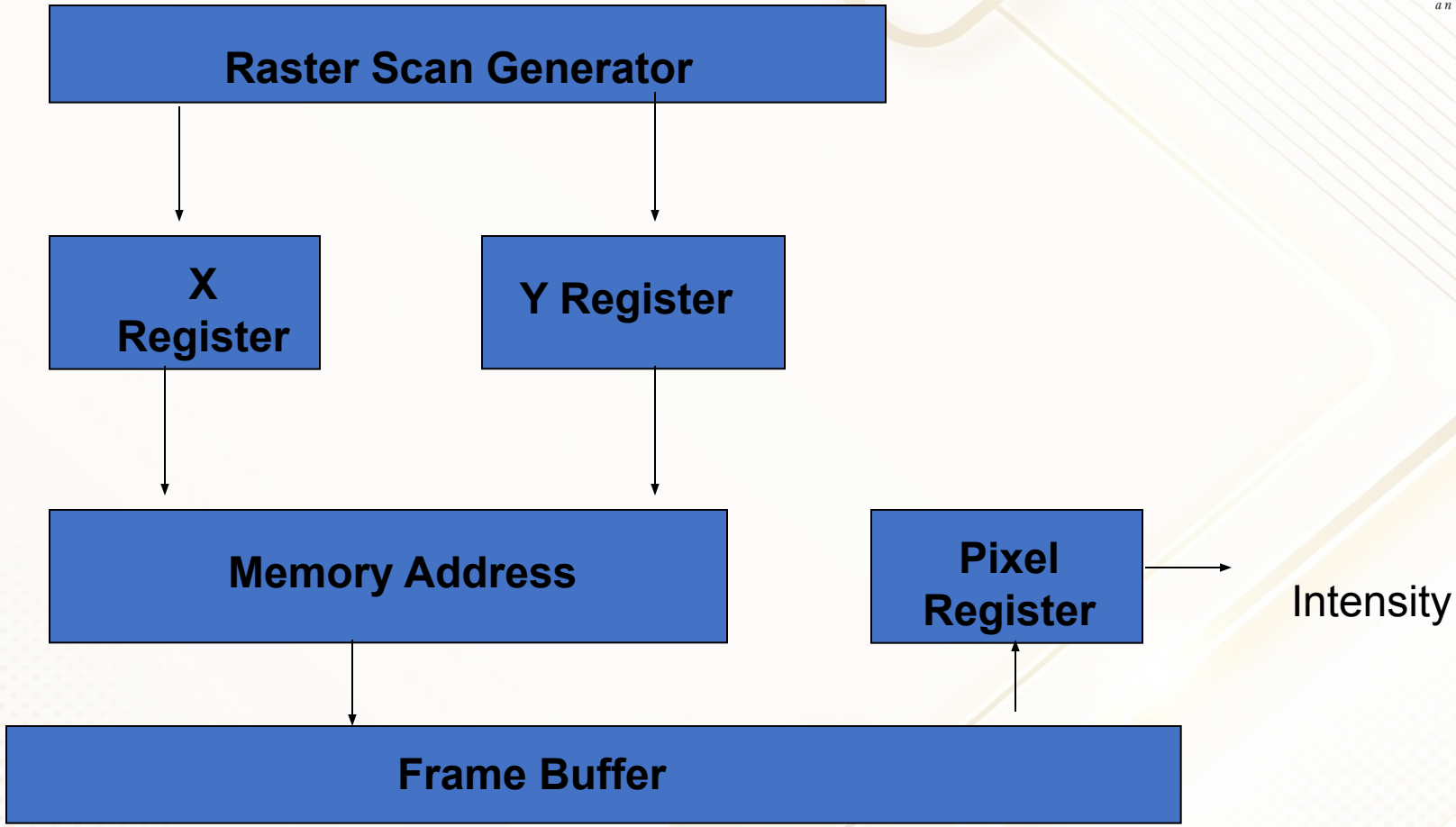
- Origin is at the lower-left corner
- We use the commands to set coordinate positions for displayed objects relative to the origin of the Cartesian reference frame.
- we can typically set the origin at any convenient location for a particular application.

x values increasing from left to the right and y values increasing from bottom to top.

Pixel positions are assigned integer x values that ranges from 0 to x_{max} and integer y values that vary from 0 to y_{max}

H/w processes reference the pixel positions from top –left corner of the screen.

Basic Video Controller Refresh Operation



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

After the last pixel on the top scan line has been processed, the x register is reset to 0 and the y register is set to value for the next scanline down.

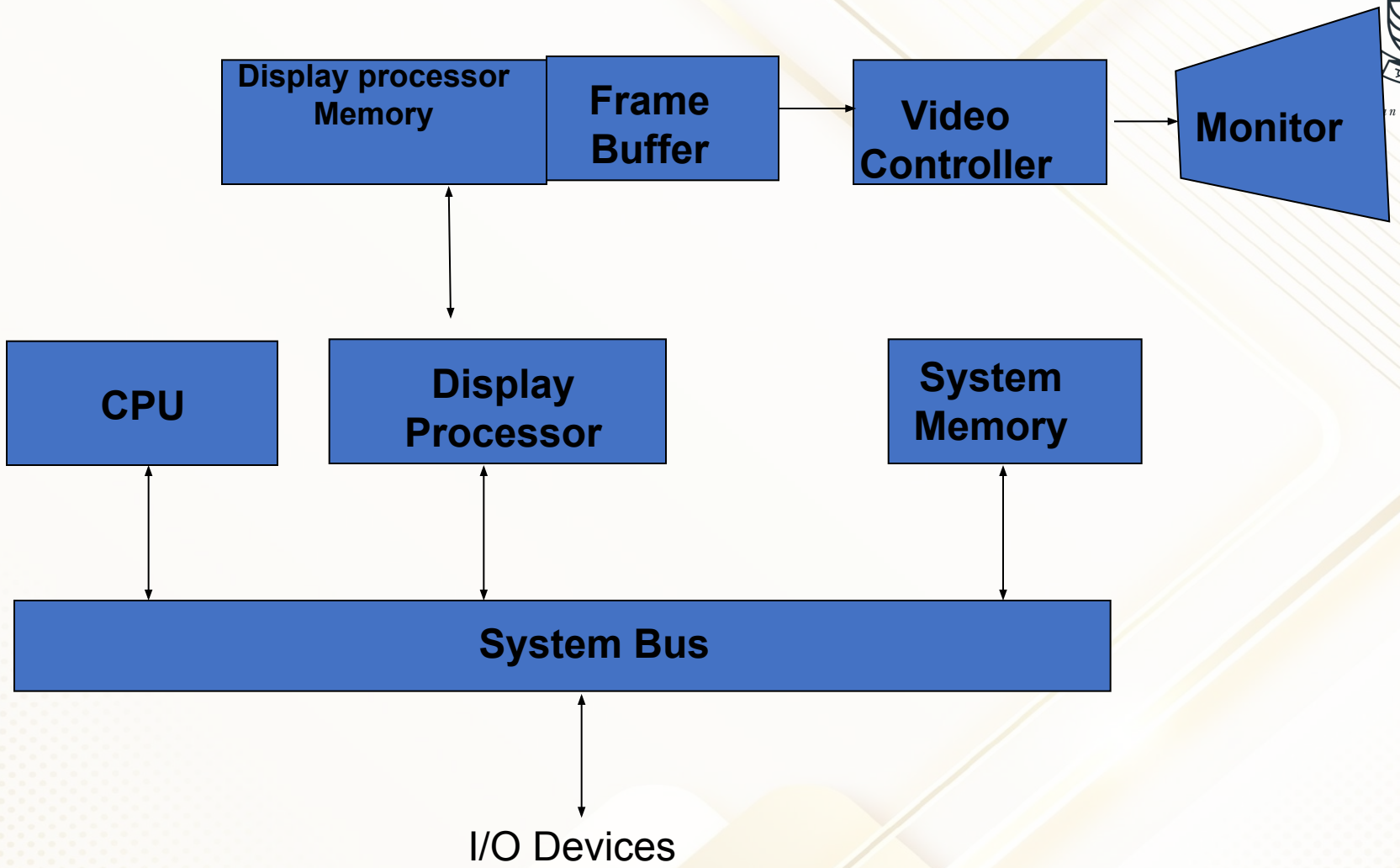
Procedure is repeated for each successive scan line.

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

To speed up pixel processing, video controllers can retrieve multiple pixel values from the refresh buffer on each pass.
The multiple pixel intensities are then stored in a separate register 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.

Other operations of video controller

retrieve pixel intensities from different memory areas on different refresh cycles.
High quality systems -two frame buffers -for generating real-time animations,
Transformations.



Architecture of Raster graphics system with a display processor

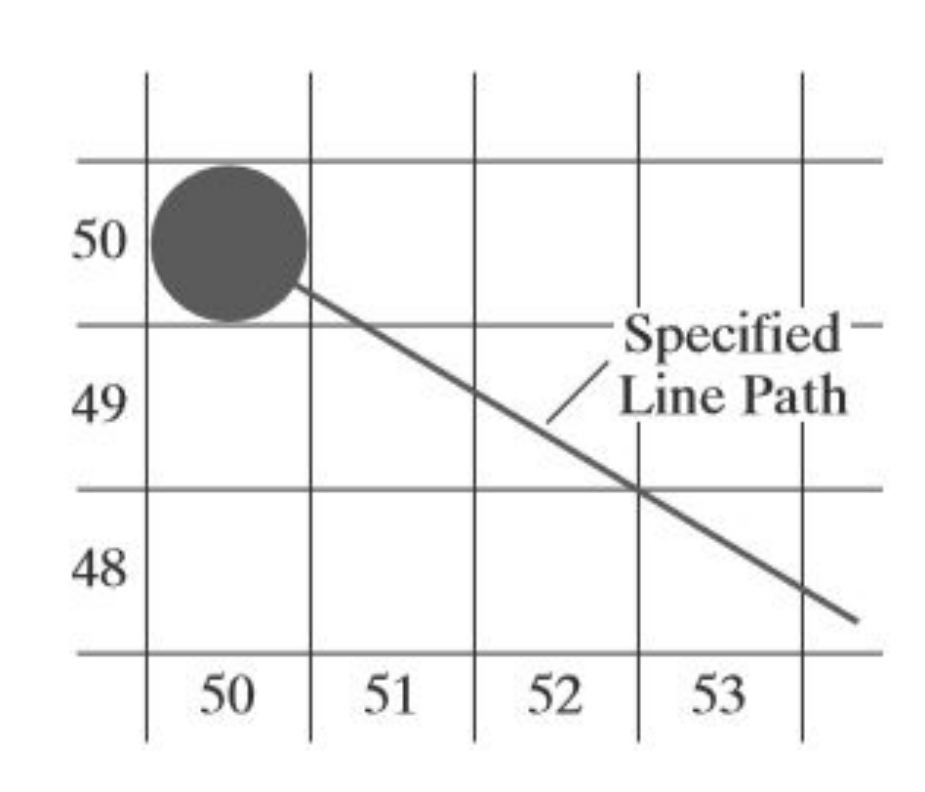
Graphics controller or a display coprocessor

- A raster system containing a separate display processor.
- Purpose
  - to free the CPU from the graphics chores.
- In addition to the system memory, a separate display processor memory area can also be provided.
- Major task
  - Digitizing a picture definition given in an application program into a set of pixel intensity values for storage in the frame buffer.
  - This digitization process is called scan conversion or rasterization

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

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.


50
49
48
Specified
Line Path
50
51
52
53

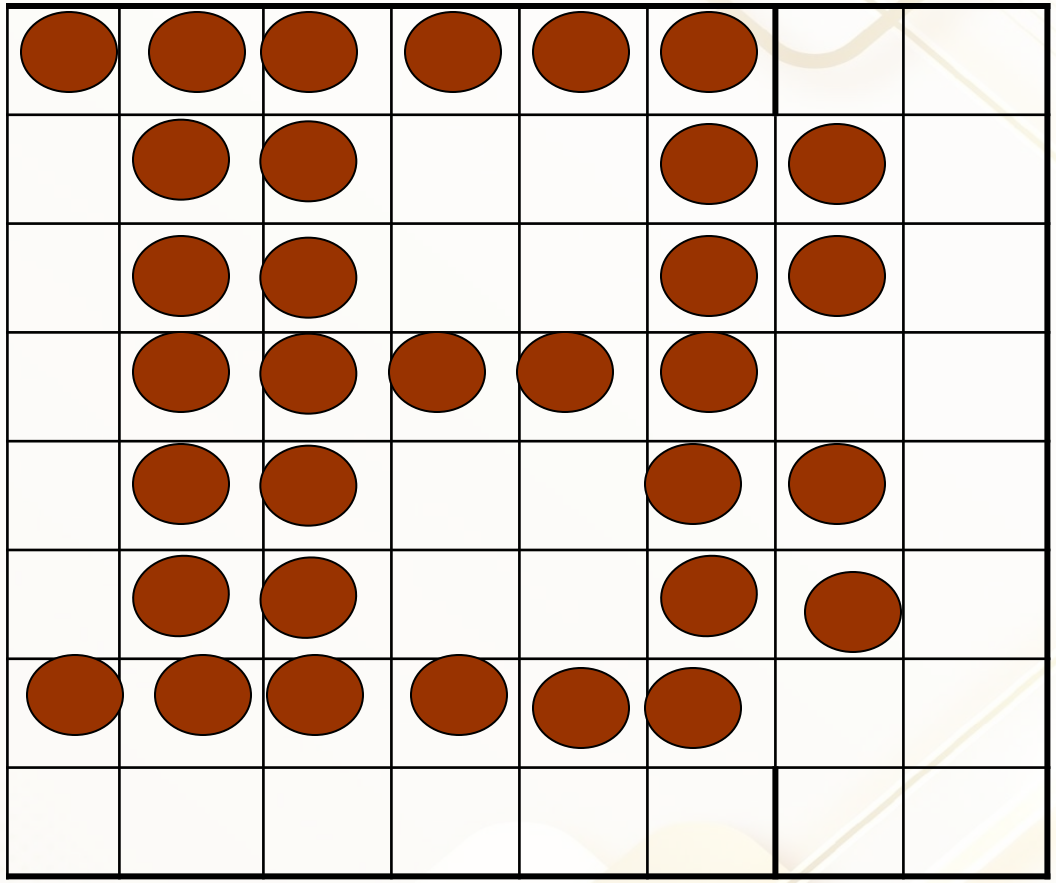
Characters

can be defined with rectangular grids

They can be defined with curved outlines.

- The array size for character grids can vary from about 5 by 7 to 9 by 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.

Raster Scan display processor



Rectangular Grid of Pixel Positions

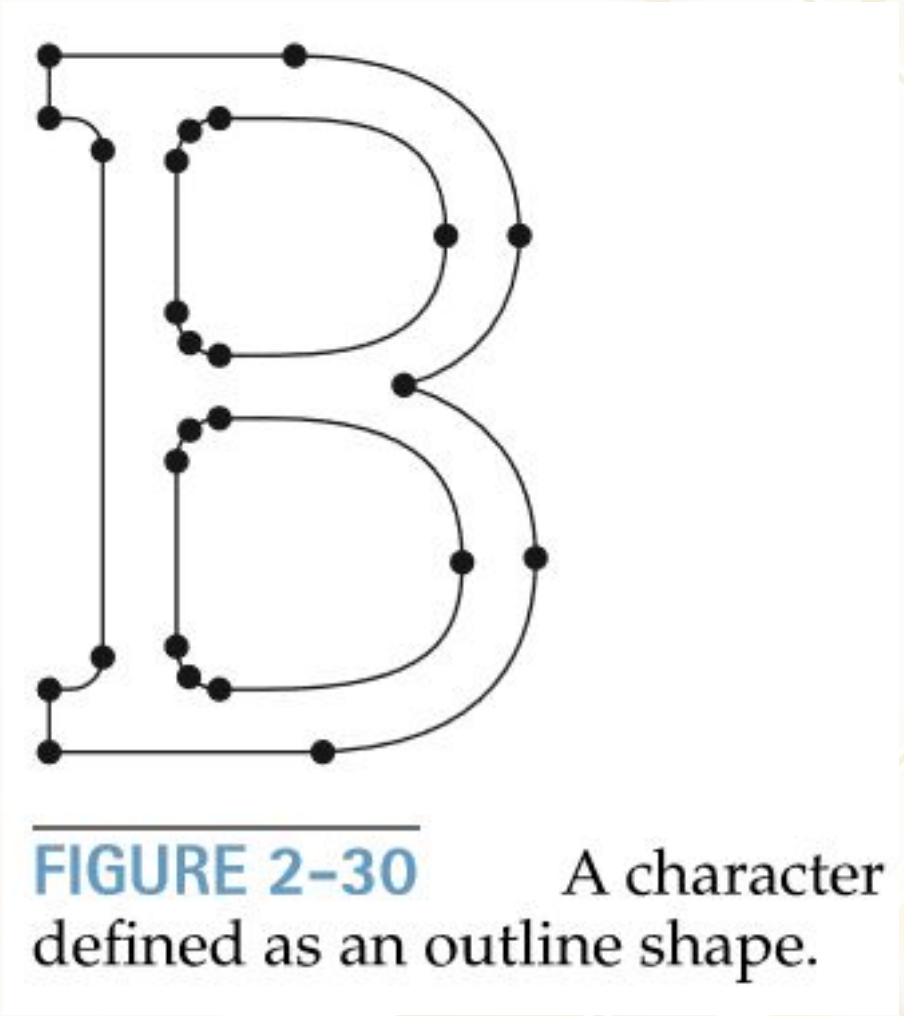


FIGURE 2-30 A character defined as an outline shape.

Additional operations of Display processors

Generating various line styles (dashed, dotted, or solid).
Displaying color areas.
Transformations and manipulations on displayed objects.
interface with interactive input devices, such as a mouse.

Graphics workstations and viewing systems

- Graphics Workstations range from **small general-purpose computer systems to multi-monitor facilities**, often with ultra-large viewing screens.

Screen resolutions for personal computer

- From about 640 by 480 to 1280 by 1024
- Diagonal screen lengths measure from 12 inches to over 21 inches.

Screen resolutions for desktop workstation

  - Vary from 1280 by 1024 to about 1600 by1200.
  - Typical screen diagonal of 18 inches or more

High-definition graphics systems, with resolutions up to 2560 by 2048, are commonly used in medical imaging, air-traffic control, simulation, and CAD.

A multi-panel display can be used to show a large view of a single scene or several individual images.

A 360◦ paneled viewing system in the NASA control-tower simulator, which is used for training and for testing ways to solve air-traffic and run way problems at airports.

FIGURE 2-33 The SGI Reality Center 2000D, featuring an ImmersaDesk R2 and displaying a large-screen stereoscopic view of pressure contours in a vascular blood-flow simulation superimposed

FIGURE 2-34 A wide-screen view of a molecular system displayed on the three-channel SGI Reality Center 3300W. (*Courtesy of Silicon Graphics, Inc. and Molecular Simulations.* © 2003

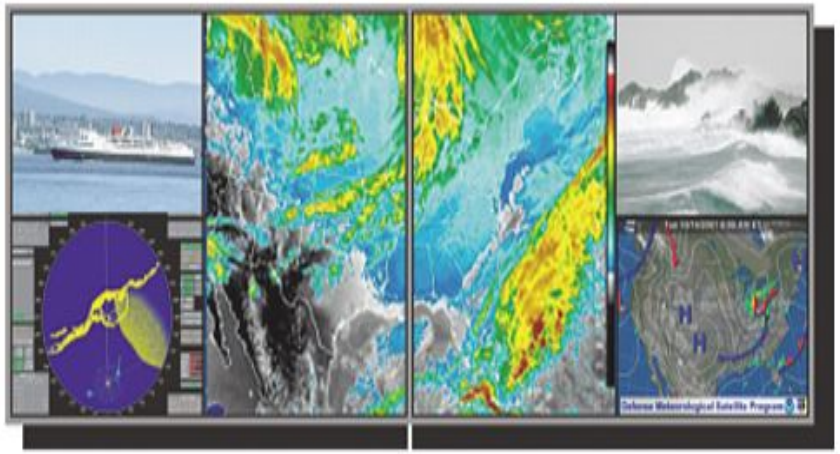


FIGURE 2-35 A multi-panel display system called the "Super Wall". (*Courtesy of RGB Spectrum.*)

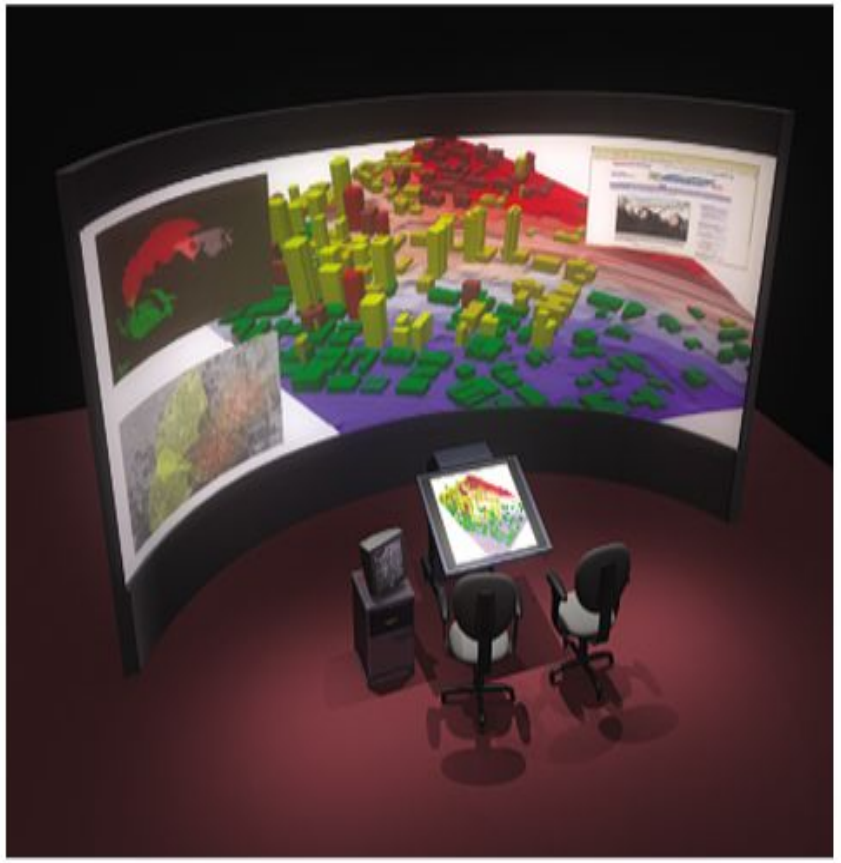


FIGURE 2-36 A homeland security study displayed using a system with a large curved viewing screen. (*Courtesy of Silicon Graphics, Inc. © 2003. All rights reserved.*)

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

360 degree screen in NASA

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Input devices

Study yourself

GRAPHICS NETWORKS

Graphics Server or server

- A graphics monitor on a network.

Client

  - The computer on the network that is executing a graphics application program
  - The output of the program is displayed on a server.
- A workstation that includes processors, as well as a monitor and input devices, can function as both a server and a client.
- collecting the instructions into packets before transmission.
- Graphics software packages often contain commands that affect packet transmission, as well as the commands for creating pictures.

GRAPHICS ON THE INTERNET

- Computers on the Internet communicate using TCP/IP
- WWW provides a hypertext system that allows users to locate and view documents that can contain text, graphics, and audio.
- Resources, such as graphics files, are identified by a *uniform resource locator (URL), that* contains two parts:

  (1) the protocol for transferring the document, and

  (2) the server that contains the document and, optionally, the location (directory) on the server

HTML

GRAPHICS SOFTWARE

Two broad classifications

- special purpose packages
- a general programming package

Special-purpose packages:

Designed for nonprogrammers, who wants to generate pictures, charts and graphs without worrying about how graphics operations work.

Example of such application packages are the artist's painting programs and various business, medical and CAD systems.

General Programming packages:

provides a library of graphics functions that can be used in a high-level programming language, such as C, C++, java, FORTRAN.

Basic functions include those for generating picture components (straight line, sphere, polygon etc),setting color and intensity values, & applying transformations.

Ex: GL,openGL, java2D, java 3D, VRML

Coordinate representations

To generate a picture, we first need to give the geometric descriptions of the objects.

These descriptions determine the locations and shapes of the objects.

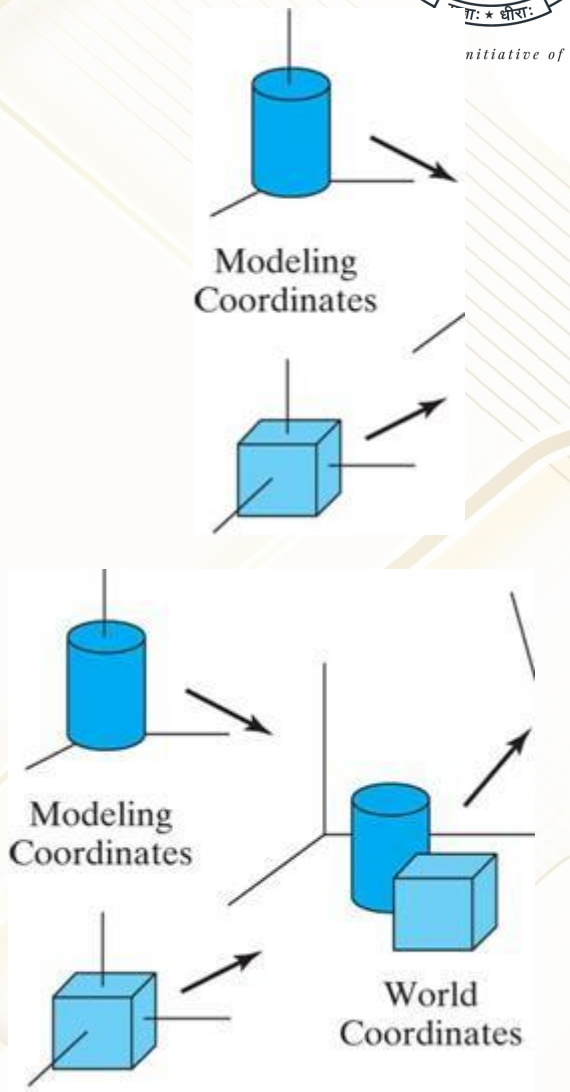
For example, a box is specified by the positions of its corners (vertices),and a sphere is defined by its center position and radius.

Graphics packages require geometric descriptions to be specified in a standard, right-handed, Cartesian-coordinate reference frame.

If coordinate values for a picture are given in some other reference frame (spherical, hyperbolic, etc.), they must be converted to Cartesian coordinates before they can be input to the graphics package.

Coordinate Representation

- **Modeling, local, master, or object coordinates:**
- Define the shapes of individual objects, such as trees or furniture, within a separate coordinate reference frame for each object.
- **World coordinates:**
- Once the individual object shapes have been specified, we can construct ("model") a scene by placing the objects into appropriate locations within a scene reference frame called world coordinates.
- This step involves the transformation of the individual modeling-coordinate frames to the world-coordinate frame.
- As an example, we could construct a bicycle …

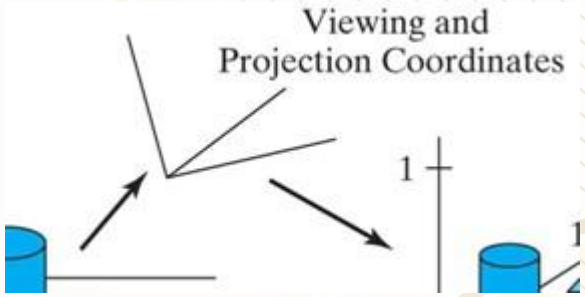


Coordinate Representation

- **Viewing pipeline:**
  - World-coordinate description is processed through various routines onto one or more output- device reference frames for display.

Viewing coordinates:

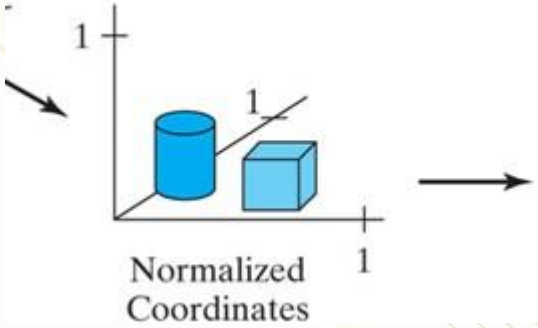
- World coordinate positions are first converted to viewing coordinates corresponding to the view we want of a scene, based on the position and orientation of a hypothetical camera.
- Then object locations are transformed to a two-dimensional projection of the scene, which corresponds to what we will see on the output device.



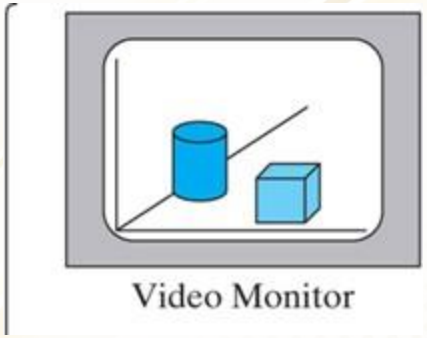
Coordinate Representation

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

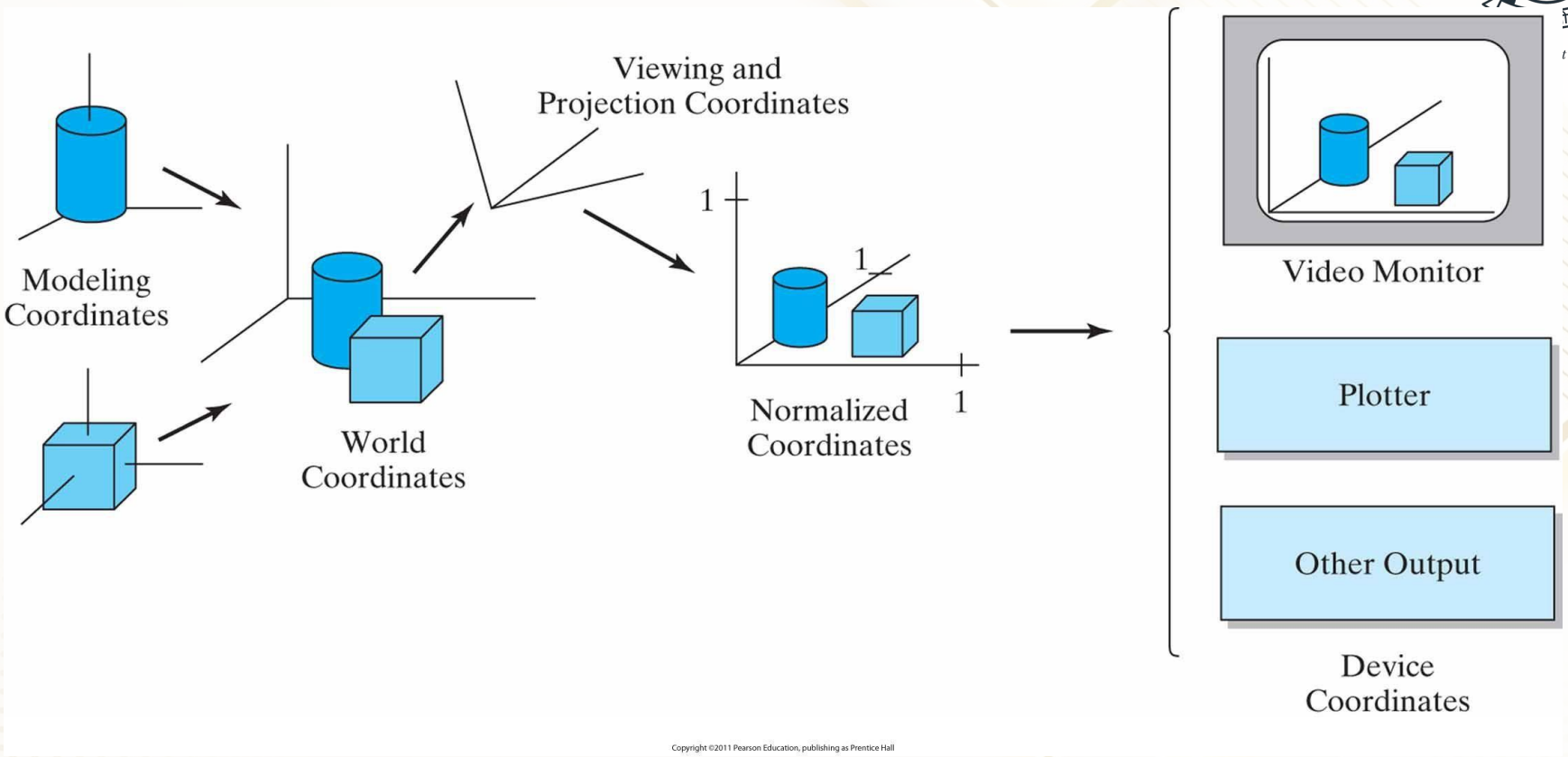
- **Normalized coordinates:**
  - The scene is then stored in normalized coordinates, where each coordinate value is in the range from -1 to 1 or in the range from **0 to 1, depending on the system.**
  - Normalized coordinates are also referred to as normalized device coordinates, since using this representation makes a graphics package independent of the coordinate range for any specific output device.



- **Device or screen coordinates**
  - Picture is scan converted into the refresh buffer of a raster system for display . The coordinate systems for display devices are generally called device coordinates, or screen coordinates in the case of a video monitor



Coordinate Representation



$$(x_{mc}, y_{mc}, z_{mc}) \rightarrow (x_{wc}, y_{wc}, z_{wc}) \rightarrow (x_{vc}, y_{vc}, z_{vc}) \rightarrow (x_{pc}, y_{pc}, z_{pc})$$
$$\rightarrow (x_{nc}, y_{nc}, z_{nc}) \rightarrow (x_{dc}, y_{dc})$$

Graphics functions

Graphics output primitives
Attributes
Geometric transformations
Modeling transformations
Viewing transformations
Input functions
Control operations

Graphics functions

- **Graphics output primitives:** The basic building blocks for pictures.
- They include character strings and geometric entities, such as points, straight lines, curved lines, filled color areas (usually polygons), and shapes defined with arrays of color points.

- Example

```
glBegin(GL_POINTS);
   glVertex2f(100.0,200.0);
glEnd();
glBegin(GL_LINES);
   glVertex2f(100.0,200.0);
   glVertex2f(300.0,400.0);
glEnd();
```

Attributes: are properties of the output primitives; that is, an attribute describes how a particular primitive is to be displayed.

- This includes color specifications, line styles, text styles, and area-filling patterns.
  - glColor3f(1.0,0.0,0.0);
  - glLineWidth(1.0);

Graphics functions

Geometric transformations: We can use to change the size, position, or orientation of an object within a scene.

- glTranslatef (tx, ty, tz);
- glRotatef (theta, vx, vy, vz);
- glScalef (sx, sy, sz);

Modeling transformations: which are used to construct a scene where individual object descriptions are given in local coordinates.

Viewing transformations: are used to select a view of the scene, the type of projection to be used, and the location on a video monitor where the view is to be displayed.

```
glViewport (xvmin, yvmin, vpWidth, vpHeight);
gluOrtho2D (xwmin, xwmax, ywmin, ywmax);
gluLookAt(…);
```

Graphics functions

- **Input functions:** Interactive graphics applications make use of various kinds of input devices, including a mouse, a tablet, or a joystick. Input functions are used to control and process the data flow from these interactive devices.

void glutKeyboardFunc(void (*func) (unsigned char key, int x, int y));

- **Control operations:** Finally, a graphics package contains a number of housekeeping tasks, such as clearing a screen display area to a selected color and initializing parameters.

```
glutInitWindowSize(500,500);
```

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Software Standards

Portability
GKS(Graphical kernal system)(2D-3D)
PHIGS (Programmers Hierarchical Interactive graphics system)
Accepted by an official national or international standard bodies by ISO and ANSI(American National Standard Institute).

GKS originally designed as a 2-D graphics packages, a 3-D GKS extension was subsequently developed.

PHIGS is a extension of GKS having increased capabilities for object modeling, color specification, surface rendering and picture manipulations.

Extension of PHIGS called PHIGS+ provide 3-D surface shading/rendering capabilities.

GL (Graphics Library)

- Silicon Graphics, Inc(SGI): Graphics Library (GL).
- De facto std.
- Fast real time rendering, extended to other h/w
- OpenGL –h/w independent version of GL.

OpenGL Architecture Review Board

- a consortium of representatives from many graphics companies and organizations.

Introduction to OpenGL

A basic library of functions is provided in OpenGL for specifying graphics primitives, attributes, geometric transformations, viewing transformations, and many other operations.
OpenGL is designed to be hardware independent, therefore many operations, such as input and output routines, are not included in the basic library. (auxiliary libraries that have been developed for OpenGL programs)

Basic OpenGL Syntax

Function names in the OpenGL basic library (OpenGL core library) are prefixed with gl, and each component word within a function name has its first letter capitalized.

glBegin, glClear, glCopyPixels, glPolygonMode

Symbolic constants begin with the uppercase letter GL. In addition, component words within a constant name are written in capital letters, and the underscore (_).

GL_2D, GL_RGB, GL_POLYGON, GL_AMBIENT_AND_DIFFUSE

Basic OpenGL Syntax

The OpenGL functions also expect specific data types.

GLbyte-8, GLshort-16, GLint-32, GLfloat-32, GLdouble-64, GLboolean-1

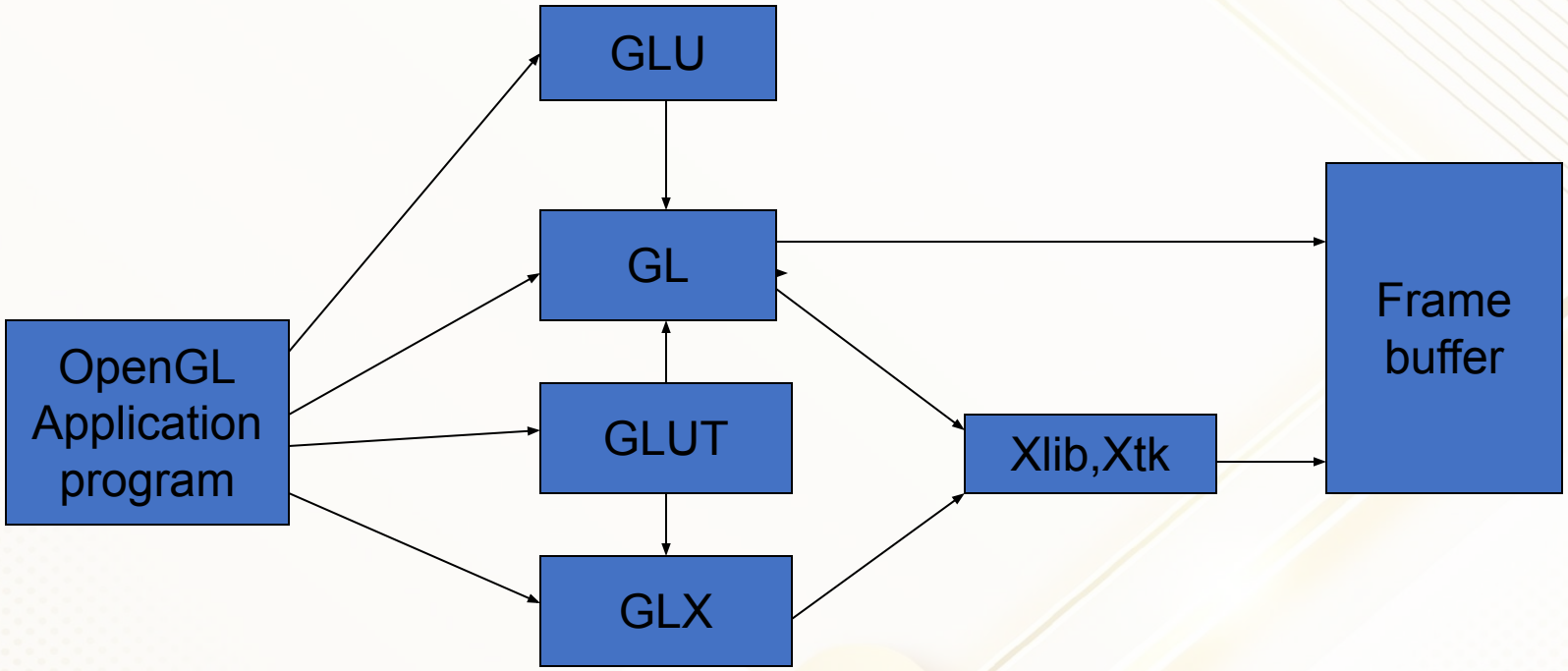
Related Libraries

The OpenGL Utility (GLU) provides routines for setting up viewing and projection matrices, describing complex objects with line and polygon approximations, displaying quadrics and B-splines using linear approximation,
all GLU function names starts with the prefix glu.

Related Libraries

- Several window-system libraries
  - The OpenGL Extension to the X window System (GLX) - the letter glx.
  - Apple systems (AGL) -prefixed with agl.
  - For Microsoft Windows systems, the WGL -prefixed with the letter wgl.
  - The Presentation Manager to OpenGL (PGL) is an interface for the IBM OS/2, which uses the prefix pgl for the library routines.
  - The OpenGL Utility Toolkit (GLUT) provides a library of functions for interacting with any screen-windowing system. The GLUT library functions are prefixed with glut.

Library organization



Related Libraries

GLUT library also contains methods for describing and rendering quadric curves and surfaces.

Since GLUT is an interface to other device specific windows systems, we can use GLUT. (device-independent)

http://www.opengl.org/resources/libraries/glut/

Header Files

In all of our graphics programs, we will need to include the header file for the OpenGL core library.
For most applications we will also need GLU.
And we need to include the header file for the window system. For WGL, windows.h header file must be listed before the OpenGL and GLU header files.

```
#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
```

Header Files

However, if we use GLUT to handle the window managing operations, we do not need to include gl.h and glu.h.

```
#include <GL/glut.h>
```

In addition, we will often need to include header files that are required by the C code.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
```

Display-Window Management Using GLUT

To get started, we can consider a simplified, minimal number of operations for displaying a picture. Since we are using the GLUT, the first step is to initialize GLUT.

```
glutInit (&argc, argv);
```

Next, we can state that a display windows is to be created on the screen with a given caption for the title bar.

```
glutCreateWindow ("An Example OpenGL Program");
```

Display-Window Management Using GLUT

- Then we need to specify what the display window is to contain. For this, we create a picture using OpenGL functions and pass the picture definition to the GLUT routine glutDisplayFunc, which assigns our picture to the display window.

  glutDisplayFunc (lineSegment);

  But the display windows is not yet on the screen. We need one more GLUT function to complete the window-processing operations.

  glutMainLoop ();

Display-Window Management Using GLUT

- glutMainLoop function must be the last one in the program. It display the initial graphics and puts the program into an infinite loop that checks for input from devices such as a mouse or keyboard.
- Although the display window that we created will be in some default location and size, we can set these parameters using additional GLUT functions.
- We use the glutInitWindowPosition function to give an initial location for the top-left corner of the display window. This position is specified in integer screen coordinates, whose origin is at the upper-left corner of the screen.

```
glutInitWindowPosition (50, 100);
```

Display-Window Management Using GLUT

Similarly, the glutInitWindowSize function is used to set the initial pixel width and height of the display window.

```
glutInitWindowSize (400, 300);
```

We can also set a number of other options for the display windows, such as buffering and a choice of color modes, with the glutInitDisplayMode function.

```
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
```

The values of the constants passed to this function are combined using a logical or operation. (single buffering and RGB, default)

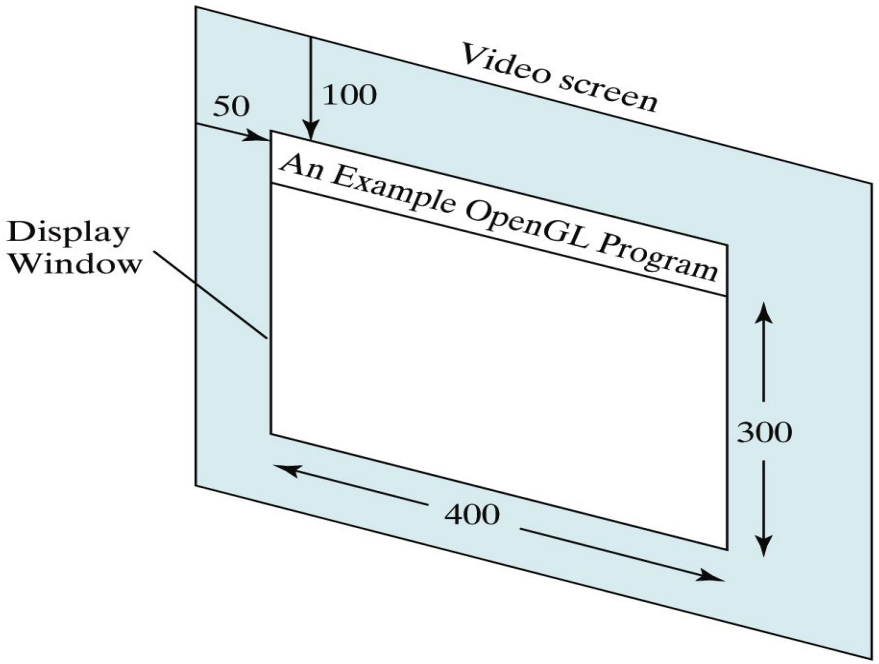


Figure 2-61

A 400 by 300 display window at position (50, 100) relative to the top-left corner of the video display.

A Complete OpenGL Program

For the display window, we can choose a background color. We need to construct a procedure that contains the appropriate OpenGL functions for the picture that we want to display.

```
glClearColor (1.0, 1.0, 1.0, 0.0);
```

The first three arguments in this function set each of the red, green, and blue component colors to the value 1.0. The fourth parameter in the glClearColor function is called the alpha value.

Alpha- 0 .0- totally transparent. 1.0- opaque

A Complete OpenGL Program

glClear (GL_COLOR_BUFFER_BIT);

The argument GL_COLOR_BUFFER_BIT is an OpenGL symbolic constant specifying that it is the bit values in the color buffer that are to be set to the values indicated in the glClearColor function.

A Complete OpenGL Program

In addition to setting the background color for the display window, we can choose a variety of color schemes for the objects we want to display in a screen.

```
glColor3f (1.0, 0.0, 0.0);
```

The suffix 3f on the glColor function indicates that we are specifying the three RGB color components using floating-point (f) value.

We simply display a two-dimensional line segment. We want to project our picture onto the display window, because generating a two-dimensional picture is treated by OpenGL as a special case of three-dimensional viewing.

A Complete OpenGL Program

```
glMatrixMode (GL_PROJECTION);
gluOrtho2D (0.0, 200.0, 0.0, 150.0);
```

- This specifies that an orthogonal projection is to be used to map the contents of a two-dimensional (2D) rectangular area of world coordinates to the screen, and that the x-coordinate values within this rectangle range from 0.0 to 200.0 with y-coordinate values ranging from 0.0 to 150.0.

- Finally, we need to call the appropriate OpenGL routines to create our line segment.

A Complete OpenGL Program

```
glBegin (GL_LINES);
        glVertex2i (180, 15);
        glVertex2i (10, 145);
glEnd ();
```

- The code defines a two-dimensional, straight-line segment with integer Cartesian endpoint coordinates (180, 15) and (10, 145).

- glFlush forces all buffers to be emptied and the OpenGL function to be processed.

```
#include<stdlib.h>
#include<stdio.h>
#include<GL/glut.h>      // (or others, depending on the system in use)

void init (void)
{
   glClearColor (1.0, 1.0, 1.0, 0.0);  // Set display-window color to white.
   glMatrixMode (GL_PROJECTION);       // Set projection parameters.
   gluOrtho2D (0.0, 100.0, 0.0, 150.0);
}
```

```
void lineSegment (void)
{
   glClear (GL_COLOR_BUFFER_BIT);  // Clear display window.
   glColor3f (1.0, 0.0, 0.0);      // Set line segment color to red.
   glBegin (GL_LINES);
      glVertex2i (180, 15);        // Specify line-segment geometry.
      glVertex2i (10, 145);
   glEnd ( );

   glFlush ( );     // Process all OpenGL routines as quickly as possible.
}
```

```
int main (int argc, char** argv)
{
   glutInit (&argc, argv);                          // Initialize GLUT.
   glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);   // Set display mode.
   glutInitWindowPosition (50, 100);   // Set top-left display-window position.
   glutInitWindowSize (400, 300);      // Set display-window width and height.
   glutCreateWindow ("OpenGL Program"); // Create display window.

   init ( );                           // Execute initialization procedure.
   glutDisplayFunc (lineSegment);      // Send graphics to display window.
   glutMainLoop ( );                   // Display everything and wait.
}
```

OpenGL Point Functions

- The basic OpenGL geometric primitives are specified by sets of vertices.

```
glBegin(type);
    glVertex*(…);
    .
    .
    glVertex*(…);
glEnd();
```

*- can be interpreted as ntv

- n- no of dimension
- t data type
- V variable are specified through a pointer to array.

For example to display point

```
glBegin(GL_POINTS);
    glVertex2f(100.0,200.0);
    glEnd();
```

To display line

```
glBegin(GL_LINES);
    glVertex2f(100.0,200.0);
   glVertex2f(200.0,300.0);
    glEnd();
```

```
int point1 [ ]= {50,100}
int point2 [ ]= {75,150}
int point3 [ ]= {100, 200}

glBegin(GL_POINTS);
 glVertex2iv(point1);
 glVertex2iv(point2);
 glVertex2iv(point3);
glEnd();
```

OpenGL Line Functions

Line segments(GL_LINES):

Successive pairs of vertices to be interpreted as the endpoints of individual segments.

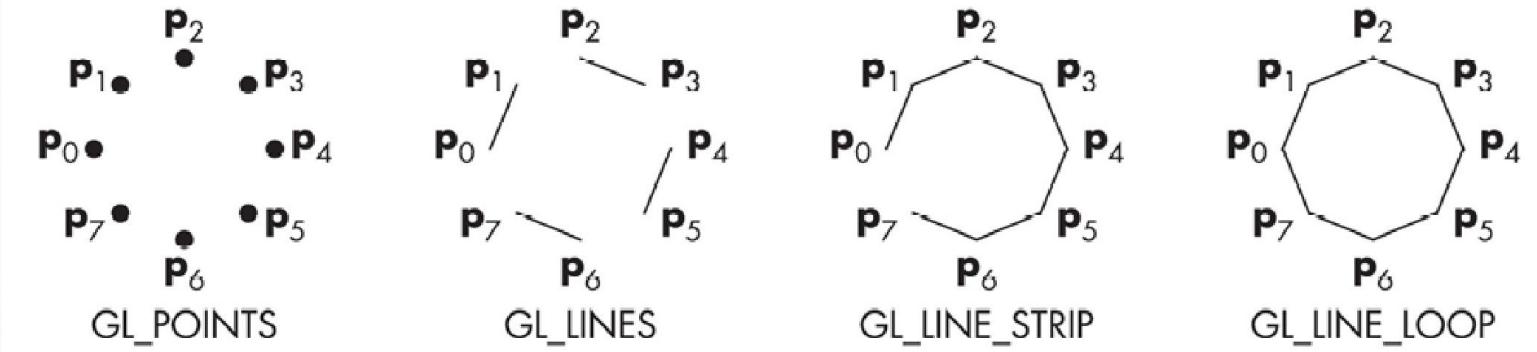
Polylines(GL_LINE_STRIP,GL_LINE_LOOP):

GL_LINE_STRIP

- If successive vertices are to be connected,we can use the line strip,or polyline form.

GL_LINE_LOOP

will draw a line segment from the final vertex to the first,thus creating a closed path.


P0
P1
P2
P3
P4
P5
P6
P7
GL_POINTS
GL_LINES
GL_LINE_STRIP
GL_LINE_LOOP

Point Attributes

Two attributes for points:

- Color and size

- Color components are set with RGB values or an Index into a color table
- Point size is an integer multiples of the pixel size.

Line attributes

Three basic attributes:
Color, width and style
Lines may be generated with other effects, such as pen and brush stokes.

Line width

- A heavy line –plot adjacent parallel lines.
- Standard-width of line- single pixels.

|m| <= 1

  - plot vertical span of pixels in each column.

- Double width- parallel line above the original line path
- Width of 3 or more- plot above and below the single-width line path

|m| > 1.0

  - Plot pixels to the right and left of the line path

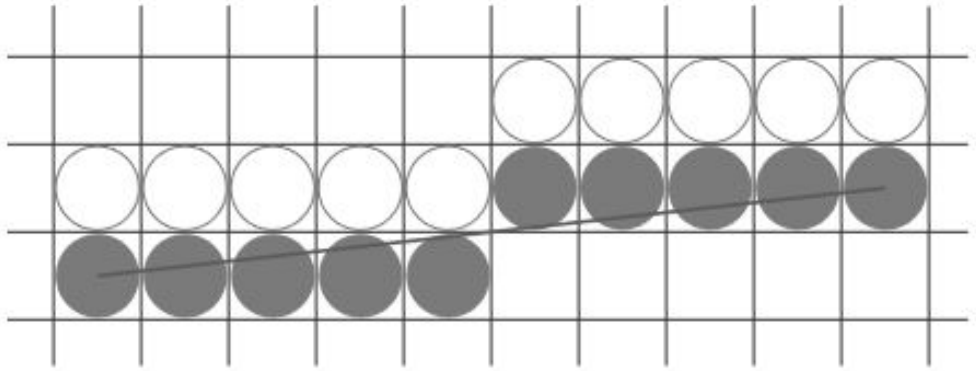


FIGURE 35
A double-wide raster line with slope $|m| < 1.0$ generated with vertical pixel spans.

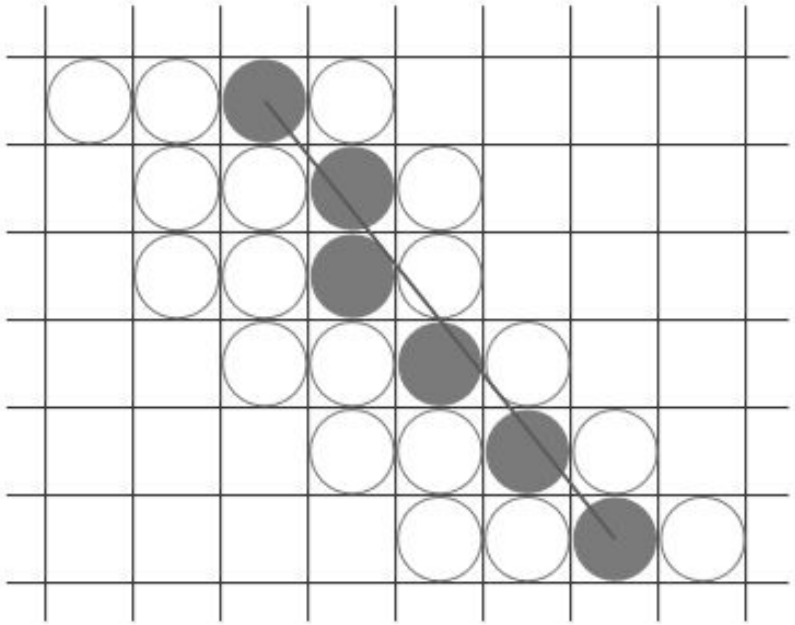


FIGURE 36
A raster line with slope $|m| > 1.0$ and a line width of 4 plotted using horizontal pixel spans.

Line Style

- Solid lines
- Dashed lines
- Dotted lines

Pixel mask

- Pixel counts for span length and inter-span spacing can be specified in a pixel mask, which is a pattern of binary digits indicating which positions to plot along line path
- 11111000- a dashed line with a dash length of five pixels and inter-dash spacing of three pixels

Unequal-length dashes for different line orientations.
Horizontal line looks small when compared to a vertical line.
Adjust the pixel counts for the solid spans and inter span spacing according to the slope of the line.

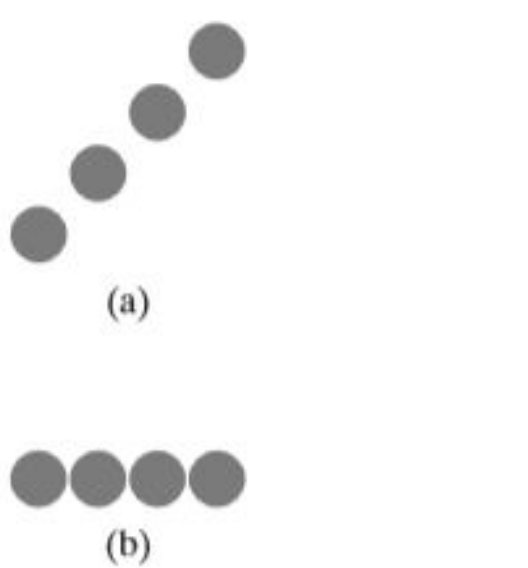


FIGURE 39
Unequal-length dashes displayed with the same number of pixels.

OpenGL Point-attribute functions

Color is specified with either **glColor** or **glIndex** function.
Set the size for an OpenGL point
glPointSize(size)
Size –positive floating point value which is rounded to an integer.
Default value of point size is 1.0

Attribute functions may be listed inside or outside of a glBegin/glEnd pair.

```
glColor3f (1.0, 0.0, 0.0);
glBegin (GL_POINTS);
glVertex2i (50, 100);
glPointSize (2.0);
glColor3f (0.0, 1.0, 0.0);
glVertex2i (75, 150);
glPointSize (3.0);
glColor3f (0.0, 0.0, 1.0);
 glVertex2i (100, 200);
glEnd ( );
```

OpenGL Line-Attribute Function

- Three basic attributes: color, width, and style.

OpenGL Line- Width Function

- Line width is set in OpenGL with the function
  - glLineWidth(width).
- Assign a floating-point value to parameter width, and this value is rounded to the nearest non negative integer.

OpenGL Line-style Function

Set a current display style for lines with the OpenGL function

- glLineStipple (repeatFactor, pattern)
  - Parameter pattern is used to reference a 16-bit integer that describes how the line should be displayed.
  - 1 bit denotes an “on” pixel position,
  - 0 bit indicates an “off” pixel position

The default pattern is 0xFFFF(each bit position has a value of 1), which produces a solid line.

Integer parameter repeat Factor specifies how many times each bit in the pattern is to be repeated before the next bit in the pattern is applied.The default repeat value is1.

Eg: 0x00FF and the repeat factor is 1.

A dashed line with eight pixels in each dash and eight pixel positions that are “off” (an eight-pixel space) between two dashes.

Start with lower order bits

Activate the line style feature of OpenGL
glEnable (GL_LINE_STIPPLE);
We can turn off the line-pattern feature with
glDisable (GL_LINE_STIPPLE);

Other OpenGL Line effects

we can vary the color along the path of a solid line by assigning a different color to each line endpoint.

Assign a blue color to one endpoint of a line and a red color to the other endpoint. The solid line is then displayed as a linear interpolation of the colors at the two endpoints:

```
glShadeModel (GL_SMOOTH);
glBegin (GL_LINES);
 glColor3f (0.0, 0.0, 1.0);
glVertex2i (50, 50);
glColor3f (1.0, 0.0, 0.0);
glVertex2i (250, 250);
glEnd ( );
```

Function glShadeModel can also be given the argument GL_FLAT.
line segment would have been displayed in a single color: the color of the second endpoint,(250,250) (red line)

Line drawing Algorithms

Define endpoints of the segment.
Determine the nearest pixel positions along the line path between the two endpoints.
Line color is loaded into the frame buffer.
Reading from the frame buffer, the video controller plots the screen pixels.
This process digitizes the line into a set of discrete integer positions that, in general, only approximates the actual line path

A computed line position of(10.48,20.51), is converted to pixel position (10, 21).
This rounding of coordinate values to Integers causes a stair-step appearance (known as“the jaggies”) in low resolution system.

Line drawing Algorithms

Line equation

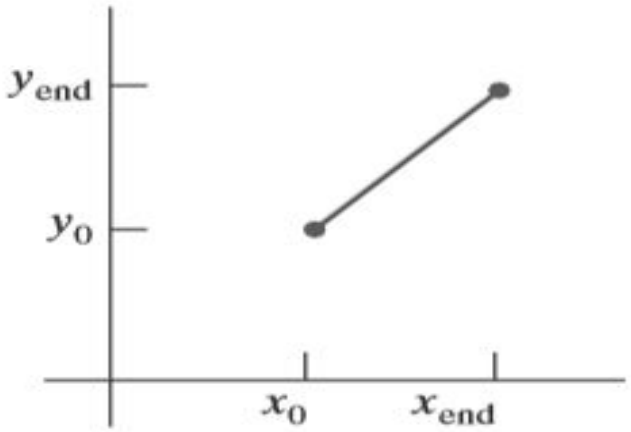


FIGURE 2
Line path between endpoint positions (x_0, y_0) and (x_{end}, y_{end}).

Line equation

- Determine pixel positions along a straight-line path from the geometric properties of the line.
- The Cartesian slope- Intercept equation for a straight line is

$$y = m \cdot x + b$$

with m as the slope of the line and b as the y intercept

- determine values for the slope m and y intercept b with the following calculations:

$$m = \frac{y_{end} - y_0}{x_{end} - x_0}$$

$$b = y_0 - m \cdot x_0$$

- For any given x interval δx along a line, we can compute the corresponding y interval, δy,

$$\delta y = m \cdot \delta x$$

- we can obtain the x interval δx corresponding to a specified δy as

$$\delta x = \delta y / m$$

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

DDA line drawing algorithm

Digital differential analyzer –scan conversion line algorithm.
Based on calculating either Δx, Δy.
Line is sampled at unit intervals in one coordinate and the corresponding integer values nearest the line path are determined for the other coordinates.

- DDA
  - Case 1: the slope is Positive and less than 1 or equal to 1
    - Sample at unit x interval ($\Delta x=1$) and compute each successive y value as :

      $y_{k+1} = y_k + m$
    - K takes integer values starting from 0,at the first point, and increasing by 1 on each step until reaching the final endpoint.
    - The calculated y must be rounded to the nearest integer.

- DDA
  - Case 2:the slope is Positive and greater than 1
    - Sample at unit y interval ($\Delta y=1$) and compute each successive y value as :

      $x_{k+1} = x_k + (1/m)$
    - K takes integer values starting from 1,at the first point, and increasing by 1 on each step until reaching the final endpoint.
    - The calculated x must be rounded to the nearest integer.

- DDA
  - Case 3:In the previous 2 cases, we start from the left to the right. If the state is reversed then:
    - If the absolute slope is less than 1, set $\Delta x=-1$ and

      $y_{k+1} = y_k - m$
  - Case 4:If the absolute slop is greater than 1, set $\Delta y=-1$ and

    $x_{k+1} = x_k - (1/m)$
  - the slope is negative, Follow above 4 cases

Line Drawing Algorithms:

- DDA properties
  - A faster method for calculating pixel positions than the direct use: y= m . x +b.
  - Uses □x or □y to eliminate the multiplication in the above equation.
  - Rounding successive additions of the floating-point increment can cause the calculated pixel positions to drift away from the true line path for long line segment.
  - Rounding and floating-point arithmetic are time-consuming operations.

DDA Algorithm

Advantage

1. Faster method for calculating pixel positions.
2. It eliminates the multiplication.

Disadvantage

1. Accumulation of round-of error.
2. Rounding operation and floating point arithmetic –still time consuming.

```
#include <stdlib.h>
#include <math.h>

inline int round (const float a)  { return int (a + 0.5); }

void lineDDA (int x0, int y0, int xEnd, int yEnd)
{
   int dx = xEnd - x0,  dy = yEnd - y0,  steps,  k;
   float xIncrement, yIncrement, x = x0, y = y0;

   if (fabs (dx) > fabs (dy))
      steps = fabs (dx);
   else
      steps = fabs (dy);
   xIncrement = float (dx) / float (steps);
   yIncrement = float (dy) / float (steps);

   setPixel (round (x), round (y));
   for (k = 0; k < steps; k++) {
      x += xIncrement;
      y += yIncrement;
      setPixel (round (x), round (y));
   }
}
```

Bresenham's Line drawing Algorithm

- An **accurate**, **efficient** raster line drawing algorithm developed by Bresenham, scan converts lines using only *incremental integer* calculations that can be adapted to display circles and other curves.

- Efficient compared to DDA algorithm because it avoid round functions.

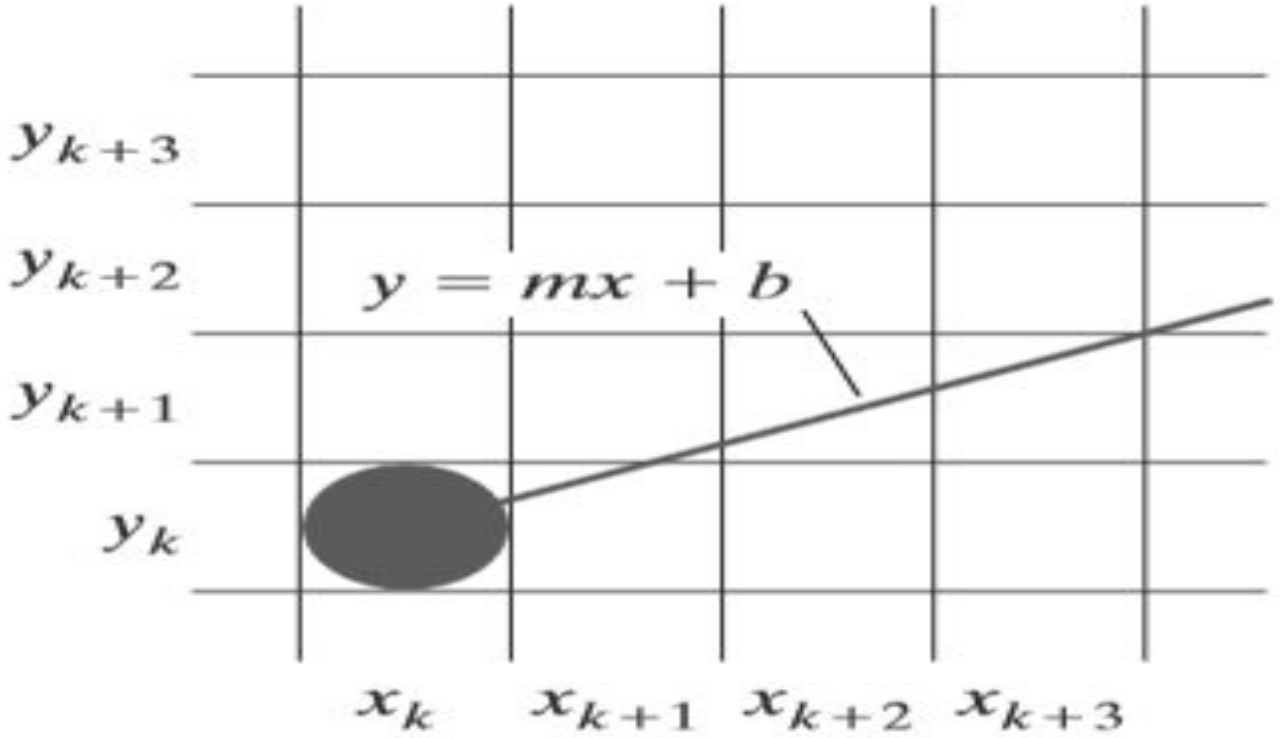


FIGURE 6
A section of the screen showing a pixel in column x_k on scan line y_k that is to be plotted along the path of a line segment with slope $0 < m < 1$.

Samples a line by incrementing by one either x or y depending on the slope of the line.

|m| < 1, increment x by 1, find y

Starting from the left end point (x_k, y_k) of a given line.

Assuming we have determined that the pixel at (x_k, y_k) is to be displayed, we next need to decide which pixel to plot in column x_k+1.

Choices are (x_k+1, y_k) and (x_k+1, y_K+1)

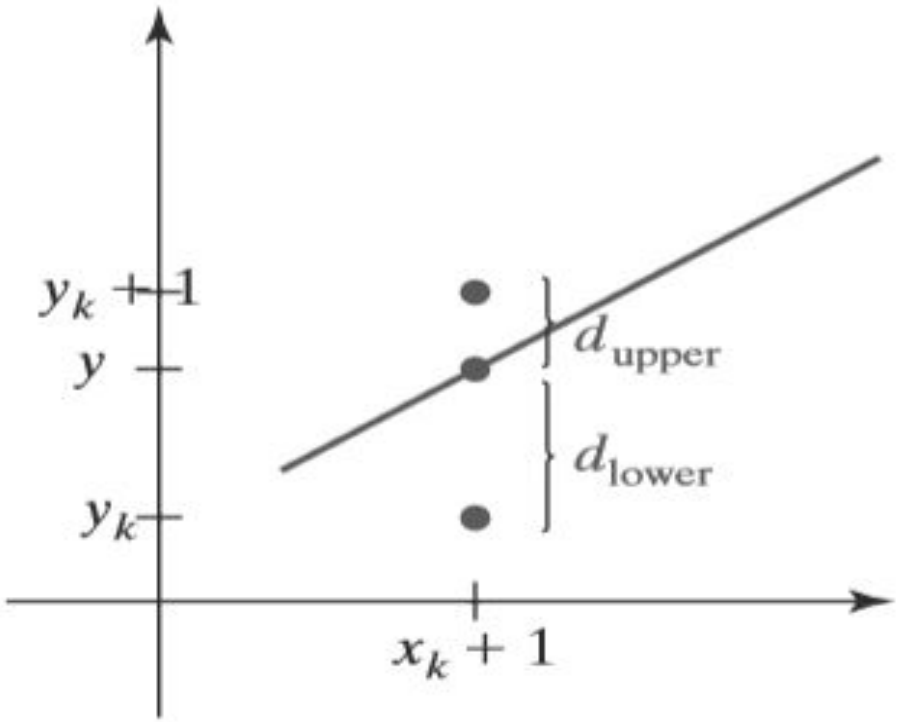


FIGURE 7
Vertical distances between pixel positions and the line y coordinate at sampling position $x_k + 1$.

- Difference between the vertical distances is computed and **the sign of the difference is used to select the pixel.**
- At sampling position x_k+1,we Label the vertical pixel separation from mathematical line path d_{lower} *and* $d_{upper.}$
- y coordinate on the math line at pixel column position x_{k+1} is calculated as
- $y=m(x_k+1)+b$
- $d_{lower}=y- y_k$
  $=m(x_k+1)+b-y_k$
- $d_{upper}= (y_k+1)-y$
  $= y_k+1-m(x_k+1)-b$

Bresenham Line Algorithm (cont)

$$d_{lower} = y - y_k = m(x_k + 1) + b - y_k$$
$$d_{upper} = (y_k + 1) - y = y_k + 1 - m(x_k + 1) - b$$

- The difference between these 2 separations is

$$d_{lower} - d_{upper} = 2m(x_k + 1) - 2y_k + 2b - 1$$

- Substitute m= *Δy/ Δx in the above equation*

- You will get $\Delta x\,(d_{lower} - d_{upper}) = 2\Delta y x_k - 2\Delta x y_k + c$
- *We call this* $\Delta x\,(d_{lower} - d_{upper})$ *as decision parametr* p_k

Bresenham's Line Algorithm

- Define

$$P_k = \Delta x\,(d_{lower} - d_{upper}) = 2\Delta y x_k - 2\,\Delta x y_k + c$$

- The sign of P_k is the same as the sign of $d_{lower} - d_{upper}$, since $\Delta x > 0$.

  Parameter c is a constant and has the value $2\Delta y + \Delta x(2b-1)$ (independent of pixel position)

- If *pixel at* y_k is closer to line-path than pixel at $y_k + 1$ *(i.e, if* $d_{lower} < d_{upper}$*)* then p_k is negative. We plot lower pixel in such a case. Otherwise , upper pixel will be plotted.
- If $p_k < 0$ then $d_{lower} < d_{upper}$ hence choose x_k+1, y_k
- If $p_k > 0$ then $d_{lower} > d_{upper}$ hence choose x_k+1, y_k+1

At step $k + 1$, the decision parameter can be evaluated as,

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

Taking the difference of p_{k+1} and p_k we get the following.

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

But, $x_{k+1} = x_k + 1$, so that

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

Where the term $y_{k+1}-y_k$ is either 0 or 1, depending on the sign of *parameter* p_k

The first parameter p_0 is directly computed

$$p_0 = 2\,\Delta y x_k - 2\,\Delta x y_k + c = 2\,\Delta y x_k - 2\,\Delta x y_k + 2\,\Delta y + \Delta x\,(2b-1)$$

Since (x_0, y_0) satisfies the line equation , we also have

$$y_0 = \Delta y / \Delta x * x_0 + b$$

Combining the above 2 equations , we will have

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

The constants $2\Delta y$ and $2\Delta y - 2\Delta x$ are calculated once for each time to be scan converted

So, the arithmetic involves only integer addition and subtraction of 2 constants

1.Input the two end points and store the left end point in (x_0, y_0)

2.Load (x_0, y_0) *into the frame buffer* ***(plot the first point)***

3.Calculate the 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 is (x_k+1, y_k) and

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

Otherwise

5.Point to plot is (x_k+1, y_k+1)

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

6.Repeat step 4 (above step) Δx -1 times

Bresenham's line algorithm properties

- It is generalized to lines with arbitrary slope by considering the symmetry between the various octants and quadrants of the xy plane.
- The constants $2\Delta y$ and $2\Delta y - 2\Delta x$ are calculated once for each line to be scan converted, so the arithmetic involves only integer addition and subtraction of these two constants.

```
#include <stdlib.h>
#include <math.h>

/*  Bresenham line-drawing procedure for |m| < 1.0.  */
void lineBres (int x0, int y0, int xEnd, int yEnd)
{
   int dx = fabs (xEnd - x0),  dy = fabs(yEnd - y0);
   int p = 2 * dy - dx;
   int twoDy = 2 * dy,  twoDyMinusDx = 2 * (dy - dx);
   int x, y;

   /* Determine which endpoint to use as start position.  */
   if (x0 > xEnd) {
      x = xEnd;
      y = yEnd;
      xEnd = x0;
   }
```

```
    else {
        x = x0;
        y = y0;
    }
    setPixel (x, y);

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

Vector Scan Display	Raster Scan Display
1. In vector scan display the beam is moved between the end points of the graphics primitives.	1. In raster scan display the beam is moved all over the screen one scan line at a time, from top bottom and then back to top.
2. Vector display flickers when the number of primitives in the buffer becomes too large.	2. In raster display, the refresh process is independent of the complexity of the image.
3. Scan conversion is not required.	3. Graphics primitives are specified in terms of their endpoints and must be scan converted into their corresponding pixels in the frame buffer.
4. Scan conversion hardware is not required.	4. Because each primitive must be scan-converted,real time dynamics is for more computational and requires separate scan conversion hardware.
5. Vector display draws a continuous and smooth lines.	5. Raster display can display mathematically smooth lines, polygons, and boundaries of curved primitives only by approximating them with pixels on the raster grid.
6. Cost is more.	6. Cost is low.
7. Vector display only draws lines and characters.	7. Raster display has ability to display areas filled with solid colours or patterns.

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Problems

1.Consider the line from (0,0) to (4,6). Digitize the line using DDA algorithm.

2.Digitize the line with endpoints (20,10) and (30,18) using Bresenham’s Algorithm