ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University

(SEWA)

SKILL ENHANCEMENT   EMPLOYABILITY   WISDOM   ACCESSIBILITY

Bachelor of Computer Applications (BCA)

Course : Computer Graphics

Course Code: BCA-4-04T

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PROGRAMME COORDINATOR :
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

PROGRAMME CO-COORDINATOR & COURSE COORDINATOR :
Dr. Gaurav Dhiman
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,

Dean Academic Affairs

Bachelor of Computer Applications (BCA)
BCA-4-04T: Computer Graphics

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

INSTRUCTIONS FOR THE PAPER SETTER/EXAMINER

1. The syllabus prescribed should be strictly adhered to.
2. The question paper will consist of three sections: A, B, and C. Sections A and B will have four questions from the respective sections of the syllabus and will carry 10 marks each. The candidates will attempt two questions from each section.
3. Section C will have fifteen short answer questions covering the entire syllabus. Each question will carry 3 marks. Candidates will attempt any ten questions from this section.
4. The examiner shall give a clear instruction to the candidates to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.
5. The duration of each paper will be three hours.

INSTRUCTIONS FOR THE CANDIDATES

Candidates are required to attempt any two questions each from the sections A and B of the question paper and any ten short questions from Section C. They have to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.

Course: Computer Graphics	
Course Code: BCA-4-04T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
CO2	Demonstrate proficiency in 2D graphics programming, including concepts like 2D transformations.
CO3	Analyze and implement key computer graphics algorithms, such as line drawing algorithms, polygon filling algorithms, and clipping algorithms.
CO4	Extract scene with different clipping methods and its transformation to graphics

	display device.
CO5	Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.

Detailed Contents:

Module	Module Name	Module Contents
Module 1	**Introduction to Computer Graphics**	Applications areas, Components of Interactive Computer Graphics System. Video Display Devices: Refresh cathode ray tube systems – raster scan CRT displays, random scan CRT displays, colour CRT-monitors, direct view storage tube. Flat panel displays – emissive vs non emissive displays, LCD displays, plasma panel displays, 3-D viewing devices, virtual reality.
Module II	**Scan conversion and 2D Graphics**	Scan converting a Point, Line (Direct, DDA and Bresenham line algorithms), Circle (Direct, Polar, Bresenham and Mid-point circle algorithms), Ellipse (Direct, Polar and Midpoint ellipse algorithms), Area filling techniques (Boundary fill, Flood fill, scan line area fill algorithm), Limitations of scan conversion. 2D Cartesian and Homogeneous co-ordinate system, Geometric transformations (Translation, Scaling, Rotation, Reflection and Shearing), Composite transformations, 2D dimensional viewing transformation and clipping (Cohen – Sutherland, Liang-Barsky, Sutherland-Hodge man algorithms).
Module III	**3D Graphics**	3D Cartesian and Homogeneous co-ordinate system, Geometric transformations (Translation, Scaling, Rotation, Reflection), Composite transformations. Mathematics of Projections: Perspective Projections - Mathematical Description and Anomalies of perspective projections. Parallel Projections – Taxonomy of Parallel Projections and their Mathematical Description. Introduction to 3D viewing pipeline and 3D clipping.
Module IV	**Hidden surface elimination algorithms**	Z-buffer, scan-line, sub-division, Painter's algorithm. Illumination Models: Diffuse reflection, Specular reflection, refracted light, texture surface patterns, Halftoning, Dithering. Surface Rendering Methods: Constant Intensity method, Gouraud Shading, Phong Shading.

Books

1. R.A. Plastock and G. Kalley, "Computer Graphics", McGraw Hill.
2. Donald Hearn and M. Pauline Baker, "Computer Graphics", Pearson Education.
3. J.D. Foley, A.V. Dam, S.K. Feiner, J.F. Hughes,. R.L Phillips, "Introduction to Computer Graphics", Addison Wesley Publishing.

Bachelor of Computer Applications (BCA)
Computer Graphics

Module I: Introduction to Computer Graphics
1.1 Applications areas.
1.2 Components of Interactive Computer Graphics System.
1.3 Video Display Devices: Refresh cathode ray tube systems – raster scan CRT displays, random scan CRT displays, colour CRT-monitors, direct view storage tube.
1.4 Flat panel displays – emissive vs non emissive displays, LCD displays, plasma panel displays, 3-D viewing devices, virtual reality.

OVERVIEW OF COMPUTER GRAPHICS

- Computer graphics deals with representation of information in pictorialform.
- It is a way of generating of images or any kind of object with the aid ofcomputers.
- It plays an important role in many fields like education, photography,filming and many more.

Definitions:

- **Computer Graphics:** It is a field of computer science that deals with creating, manipulating, storing and displaying pictures or objects on a display device.
- **Pixel:** It is a smallest element on the display screen.
- **Resolution:** The number of pixels on a display monitor expressed in terms of number of pixels on horizontal axis on vertical axis is called as resolution.
- **Rasterization:** The process of determining appropriate pixels for representing picture or object on screen is called as rasterization.
- **Scan conversion:** The process of representing continuous picture or object as a collection of discrete pixel is called as scan conversion.

Advantages of Computer Graphics:

- Produces high quality graphics.
- Provides use of tools to generate and manipulate objects or pictures.
- Used to work with 2D and 3D objects.
- Used in animations.

COMPUTER GRAPHICS APPLICATIONS

- Computer graphics can be used in various fields. Some of the common applications are Computer Aided Design, Computer Art, Entertainment and Education and Training

Computer Aided Design:

- Design processes of architecture and engineering systems.
- Design of automobiles, textiles, spacecraft, aircrafts, buildings, etc.

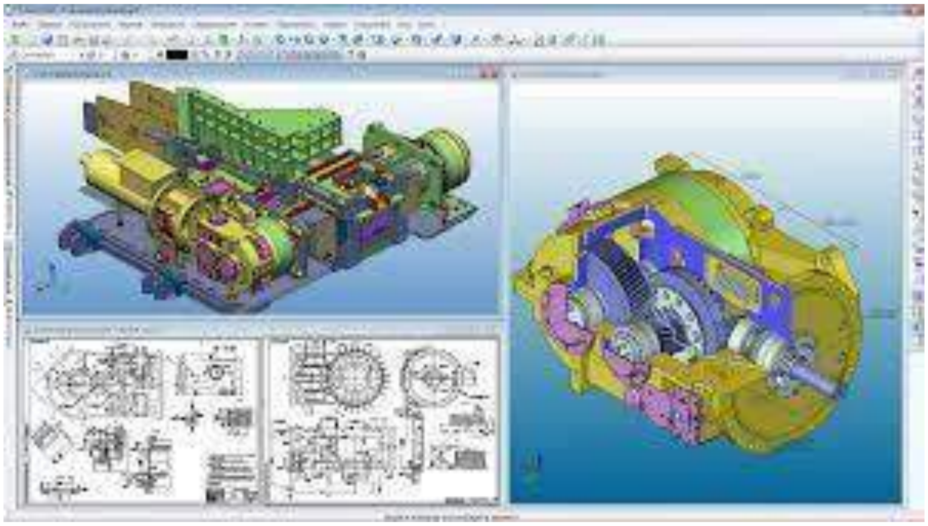


Fig. 1 Use of CAD in computer graphics

Computer Art:

- It is majorly used in the field of fine arts and commercial applications.
- Use of applications like mathematics package, paint package, animation program and desktop publishing software helps in computer art. E.g., cartoon drawing, painting.

Fig. 2 Use of Computer Art in computer graphics

Entertainment:

- It can be used to create any motion pictures and videos.
- Animated movies or video clips can be created.
- Sound and music can be used to generate any movie clips, animatedshows or TV shows.
- Games can also be made with interactivity features.

Fig. 3 Use of Entertainment in computer graphics

Education and Training :

- It is used in classroom to enhance educational experience by simplifying teacher's work.
- Models can be generated which can be useful for teaching and making understanding more easier.
- In distance learning, where students are not at the same place, use of computer graphics plays an important role, by means of conducting tests. E.g., kahoot.it, quizzes.com.
- For training, simulators can be made as hands on experience for candidates for better understanding.

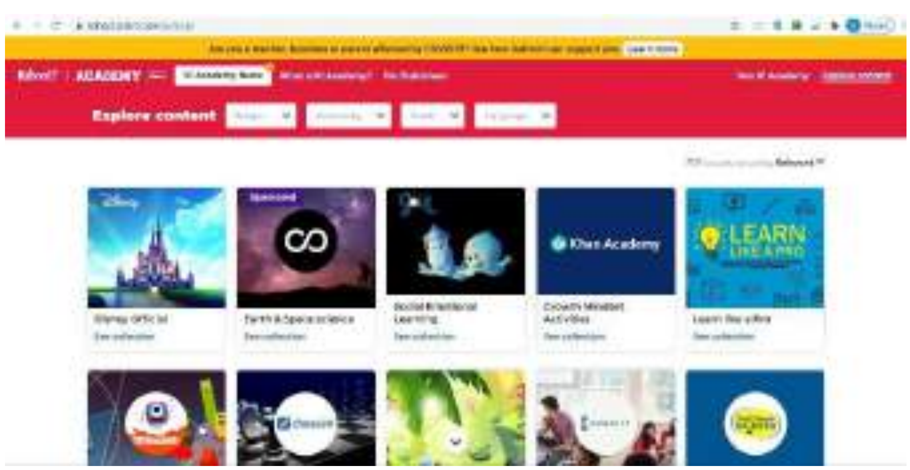


Fig. 4 Use of Kahoot in Education

Presentation Graphics:

- Reports can be generated on slides.
- It is used for summarizing mathematical, financial, statistical andscientific data.
- Graphs and charts can be generated to make any kind of presentation.E.g., pie charts, bar charts.

Fig. 5 Use of Presentation graphics in computer graphics

Visualization:

- To check insights of data, to study behavior of processes, visualizationis required with proper use of computer graphics.

Fig. 6 Use of Visualization in computer graphics

Image Processing:

- Used in applications where images or photographs requires editing.
- Processing on existing images done for more enhancements.

Fig. 7 Use of Image Processing in computer graphics

Graphical User Interfaces :

- A window manager is an important component in GUI that allows auser to display multimedia window areas.
- For fast selection processing, use of menus, icons and buttons are madein an interface.

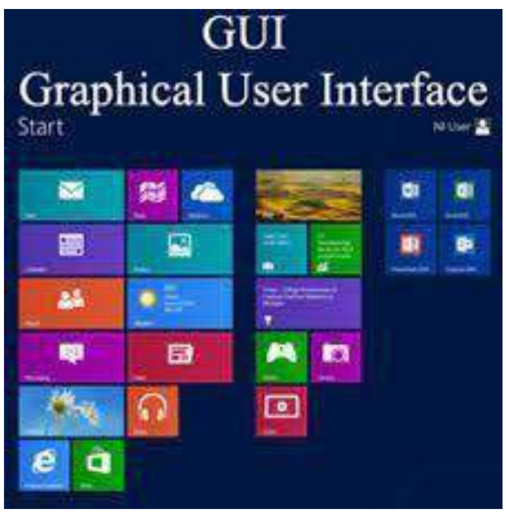


Fig. 8 Use of GUI in computer graphics

COMPUTER GRAPHICS SOFTWARE

- Graphics software can be classified into two categories:

a. General Programming Packages

b. Special Purpose Applications Packages

a. General Programming Packages

- They are like programming languages; they contain rich set of graphics functions.
- Basic functions include:
  - Generate picture components like circles, polygons, straight lines, etc.

    - Filling colors and setting intensity values
    - Selecting views
    - Applying transformations
  - E.g., graphics library system, Open GL, Open CV

b. Special Purpose Application Packages

  - These set of graphics packages are designed for non-programmers.
  - Users can create their own graphics without worrying about the underlying concepts.
  - The interface to the graphics routines in such packages allows users to communicate with the programs in their own terms.
  - E.g., Paint, CAD, etc.

Coordinate Representation:

- General graphics packages are generally designed to be used with Cartesian coordinate specifications.
- The coordinate values for an object should be converted to Cartesian coordinates if they were specified in some other reference frame, before they are passed as input to a graphics package.
- Special purpose packages allow the use of other coordinate systems based upon the applications used.
- Various Cartesian reference frames can be used to construct and displaya scene.
- Individual shape of an object can be constructed in a scene within separate coordinate reference frames called modelling coordinates or local coordinates or master coordinates.
- Objects can be placed into appropriate positions within a scene using a reference frame called as world coordinates, once the individual object shapes have been specified.
- For display, the world coordinate description of the scene is transferred to one or more output device reference frames, called as device coordinates or screen coordinates.
- Before final conversion to specific device coordinates, a graphics system first converts world coordinates positions to normalized device coordinates in the range from 0 to 1.
- This is done to ensure that the system is independent of any device usedas a workstation.

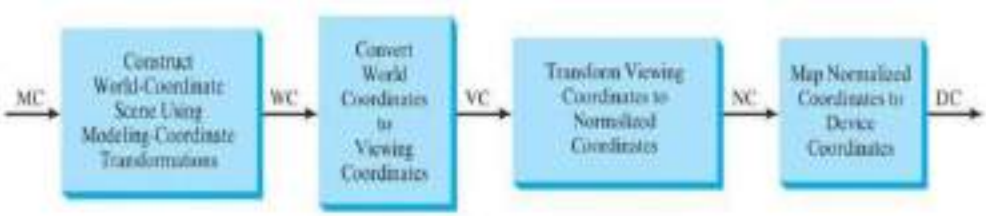


Fig. 9 Viewing Pipeline

Graphics Functions:

- Users are provided with various functions for creating and manipulating pictures by general-purpose graphics package.
- **Output primitives:** These are the basic building blocks for pictures. It includes string and entities like points, straight lines, curved lines,circles and polygons.
- **Attributes:** The properties of output primitives are referred to as attributes that describes how a primitive is to be displayed. It includes line styles, color specifications, intensity of colors and text styles.
- **Geometric transformations:** It is used to change the size, orientationor position of an object within a scene.
- **Viewing transformations:** It is used to specify view that is to be presented and portion of the output display area that is to be used.
- **Input functions:** It is used to control and process the data flow from interactive devices.
- **Control operations:** It is used in clearing a display screen or initializing parameters.

Software Standards:

- The primary intension of standardization of graphics software is portability.
- The packages are designed with standard graphics functions so they can be used with different hardware systems and used in different implantation and applications.
- The **Graphical Kernel System (GKS)** was developed to adopt the firstgraphics software standard by the International Standard Organization (ISO).
- Another graphics standard developed and approved was **Programmers Hierarchical Interactive Graphics System (PHIGS)**, which is an extension of GKS.
- It provides improved capabilities for object modelling, color specifications, surface rendering and picture manipulation.
- An extension of PHIGS, called as PHIGS+, was developed to provide 3D surface shading capabilities.
- PHIGS does not provide a standard methodology for graphics interface to output devices and does not specify methods for storing and transmitting pictures.
- **Computer Graphics Interface (CGI):** system provides

standards for device interface methods and the **Computer Graphics Metafile (CGM)** system specifies standards for archiving and transporting pictures.

INPUT DEVICES

Keyboards:

- To enter any kind of string or number, an alphanumeric keyboard isused as an input in a graphics system.
- A general keyboard can consist of 84 keys or 101 keys.
- A keyboard consists of numeric keys (0 to 9), alphabetic keys (A to Z), control keys (Ctrl, Shift, Alt), special symbol keys (!, @, #,", ?), cursor control keys (arrow keys), function keys (F1 to F9).
- Keyboards are not only used to enter any numbers, or symbols or alphabets, it is also be used for menu selection.
- A numeric keypad is on the right of keyboard for fast entry of numeric data.

Fig. 10 Keyboard

Mouse:

- To position the cursor on the screen, mouse is used.
- The roller present at the bottom of the mouse records the direction ofthe movement.
- The motion of the mouse can also be detected by option sensor in which the mouse is moved over a special mouse pad that has a grid of horizontal and vertical lines.
- The optical sensor detects movement across the lines in the grid.
- The mouse consists of two or three buttons on the top to execute any kind of operation.
- There is a type of mouse called as Z mouse which can be used to pick an object, rotate that object, move the object in any direction and navigate our view through a three dimensional scene.

Fig. 11 Mouse

Trackball and Spaceball:

- A trackball is a ball that is used to rotate with palm of the hand or with the fingers, to produce screen-cursor movement.
- A resistor with sliding or rotating contact (called potentiometers) attached to the ball, is used to measure the amount and direction of rotation.
- Trackballs are generally mounted on keyboards and is a two-dimensional positioning device.
- A spaceball does not actually move.
- A sensor whose resistance varies when force applied (called strain gauges) measures 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 CAD, animation, virtual-reality systems, modeling and other applications.

Fig. 12 Trackball and Spaceball

Joysticks:

- A joystick consists of a small, vertical lever (called the stick) mountedon a base that is used to move the screen cursor in all directions.
- Position on the screen is selected with actual stick movement or byapplying pressure on stick.
- Joysticks can be standalone device or can be mounted on a keyboard.
- The direction of the screen-cursor movement depends upon the direction in which the stick is moved.
- Potentiometers mounted at the base of the joystick measure the amount of movement, and springs return the stick to the center position when it is released.

Fig. 13 Joystick

Data Glove:

- Data glove is used to grasp a virtual object.
- The glove is made up of many sensors that helps to detect motions of fingers and hand.
- The information about the position and orientation of hand is provided by the electromagnetic coupling between transmitting and receiving antennas.
- 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.

Fig. 14 Data Glove

Image Scanner :

- 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.
- Once we have the internal representation of a picture, we can apply transformations to rotate, scale, or crop the picture to a particular screenarea.
- For scanned text input, various editing operations can be performed onthe stored documents.
- Some scanners are able to scan either graphical representations or text,and they come in a variety of sizes and capabilities.

Fig. 15 Image Scanner

Touch Panels :

- A type of display screen that has a touch-sensitive transparent panelcovering the screen are called touch panels.
- The input is recorded through finger when in contact with the screen.
- It allows a user to interact directly what is being displayed instead ofusing mouse and keyboards.
- Generally used on smart phones and laptops.

Fig. 16 Touch Panel

Light Pens:

- It is a pencil-shaped device used to select screen positions by detectingthe light coming from points on the CRT screen.
- Also used to select menu driven applications or draw pictures onmonitor screen.
- It consist of photocell and optical system placed in a small tube.
- When its tip is moved over the monitor screen, the photocell-sensing element detects screen location and sends corresponding signals toCPU.

Fig. 17 Light Pens

Voice Systems:

- It is used in some graphics workstations as input devices to acceptvoice commands.
- These systems operate by matching an input against a predefined dictionary of words and phrases.
- It is easy to use and most efficient device E.g., a microphone.

Fig. 18 Voice Systems

DISPLAY DEVICES

- There are number of display devices used in graphics.
- Output devices is another name of display devices.
- The display devices generally used are CRT and LCD.

Storage Tubes Graphics Displays:

- This is also known as Direct View Storage Tubes (DVST).
- The raster scan display refreshes the screen to maintain a screen image whereas DVST gives alternate method of maintaining the screen image without refreshing.

Components of DVST:

i) Two Electron Guns

ii) Phosphor coated screen,

iii) Storage grid mesh

iv) Collector

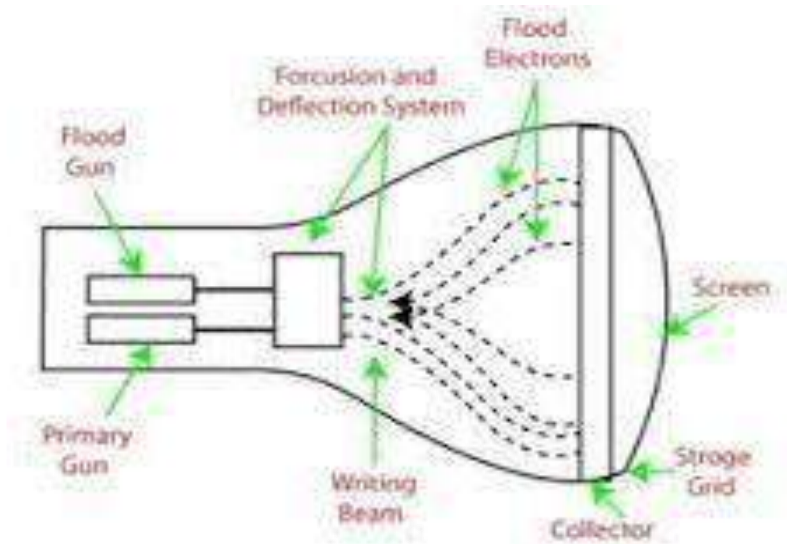


Fig 19. Working of Direct View Storage

Tube **Working Principle:**

1. There are two electron guns, namely primary gun and flood gun.
2. The primary gun stores the picture pattern and the flood gun maintains the picture display.
3. A primary gun produces high-speed electrons, which strike on the storage grid to draw the picture pattern.
4. Continuous low speed electrons from flood gun pass through the control grid and got attracted to the positive charged areas of the

storage grid.

5. The low speed electron then penetrates the storage grid and strike the phosphor coating.
6. During this process, the collector behind the storage grid smooths out the flows of food electrons.

Merits:

- It has flat screen.
- Refreshing of CRT is not required.
- Complex picture can be displayed on high resolution.

Demerits:

- The performance of DVST is poor compared to CRT.
- Erasing requires removal of charge on the storage grid.
- Selective or part erasing of screen is not possible.
- Erasing of screen produces flash over the entire screen.

Calligraphic Refresh Graphics Displays:

Calligraphic refresh graphic displays are also referred as random scan or stroke-writing or vector displays.

Components:

A CRT display unit with electron beam,, Refresh display files,Graphics controller

Working Principle:

- A CRT, as a random scan display unit, has an electron beam directedonly to the parts of the screen where a picture is to be drawn.
- Random scan monitors draw a picture one line at a time.
- The component of a picture can be drawn and refreshed by a random scan system in a specified order.
- A picture definition is stored as a set of line-drawing commands in an area of memory called a refresh display file or refresh buffer.
- To display a specified picture, the system cycles through the set of commands in the display file, drawing each component line one by one.
- After all line drawing commands have been processed, the system cycles back to the first line command in the list and repeat the procedure of scan, display, and retrace.

Example:

- A pen plotter

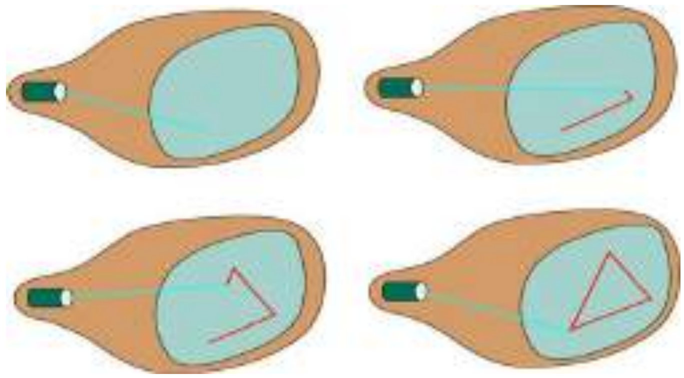


Fig 20. Working of Random Scan Systems

- Random scan displays draw all the component lines of a picture with 30to 60 times for each second.
- High quality vector systems are capable of handling approximately 100,000 short lines at this refresh rate.
- The faster refreshing of the set of lines could burn out the phosphor. Therefore, when a small set of lines are to be displayed, each refresh cycle is delayed to avoid greater refresh rates, typically 60 frames per second.
- Random scan systems are designed for line drawing applications; henceit cannot display realistic shaded scenes.
- Vector displays have higher resolution than raster systems, as picture definition is stored as a set of line drawing instructions instead of a set of intensity values for all screen points.
- These vector displays produce smooth line drawings, because the CRT beam directly follows the line path.

Raster Refresh Graphics Displays:

A raster scan display is the most common method of drawing images on a CRT screen.

Working Principle:

- In this method, horizontal and vertical deflection signals are generated to move a beam all over the screen in a pattern for displaying anyimage.
- The electron beam is swept across the screen one row at a time fromtop to bottom.
- The electron beam sweeps back and forth from left to right across the screen.

Horizontal Retrace

While electron beam moves from left to right the beam is on and when it moves back from right to left the beam is off. This phenomenon is known as horizontal retrace.

Vertical Retrace

As soon as the beam reaches the bottom of the screen, it is turned off and is rapidly retraced back to the top to start again. This is known as vertical retrace.

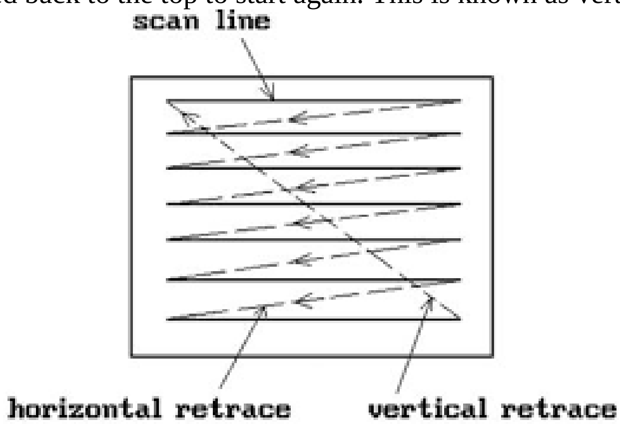


Fig 21. Horizontal and vertical retra

- Raster scan displays maintain the steady image on the screen by repeating scanning of the scan image. This process is known as refreshing the screen.

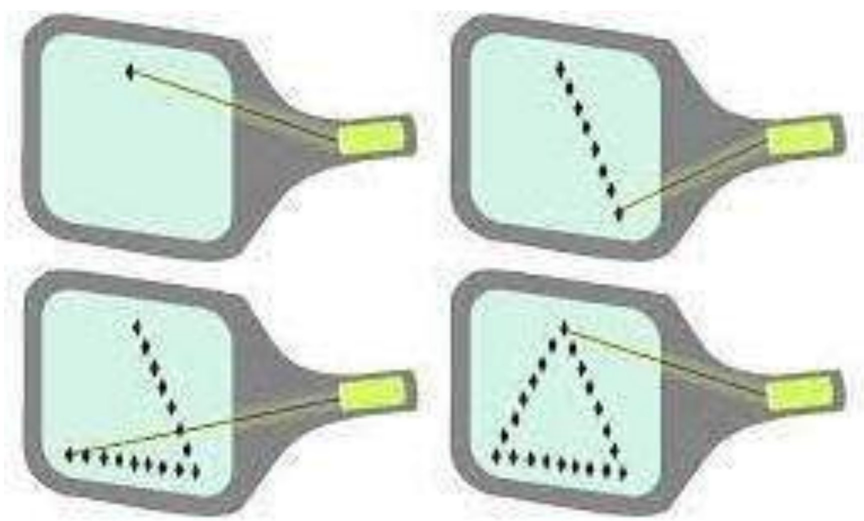


Fig 22. Working of Raster Scan Systems

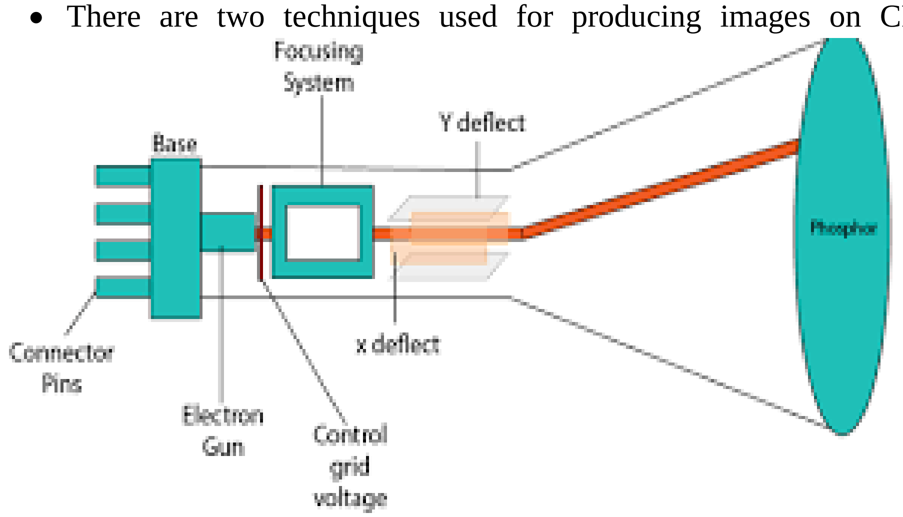
Random Scan Display V/S Raster Scan Display:

Random Scan Display	Raster Scan Display
It has high resolution.	It has less resolution.
The electron beam is directed on portion of screen where picture is to be displayed.	The electron beam is directed to the whole screen i.e. scanning one row at a time.
It is expensive.	It is less expensive.

It stores picture definition as a set of line commands called refresh display file.	It stores picture definition in refresh buffer or frame buffer.
Smooth line is produced.	Zigzag line is produced.
Hard to display realistic image.	Realistic image is displayed.
It uses line drawing applications.	It uses pixels along scan line to draw an image.
Refresh rate is 30 to 60 frames/second.	Refresh rate is 60 to 80 frames/second.
Memory required is less.	Memory required is more.

Cathode Ray Tube (CRT) :

- A CRT is an evacuated glass tube.
- An electronic gun at the rear of the tube produces a beam of electron, which is directed towards the front of the screen by a high voltage.
- The inner side of the screen is coated with phosphor substance, which gives off light when it is stroked by electron.
- The control grid voltage determines the velocity achieved by the electrons before they hit the phosphor.
- The control grid voltage determines how many electrons are actually in the electron beam.
- Thus, control grid controls the intensity of the spot where the electron beam strikes the screen.
- The focusing system concentrates the electron beam so that the beam converges to a small point when it hits the phosphor.
- The deflection system of CRT consists of two pairs of parallel plates, namely horizontal and vertical deflection plates.
- The voltage applied to vertical deflection of the electron beam and the voltage applied to the horizontal deflection plates controls the horizontal deflection of the electron beam.
- There are two techniques used for producing images on CRT



Cathode Ray Tube

screen, namely Random Scan and Raster Scan.

Fig 23. Working of CRT Monitors

Color CRT:

- A CRT monitor displays color pictures by using a combination of phosphors that emit different-colored light.
- It generates a range of colors by combining the emitted light from different phosphors.
- There are two basic techniques used for producing color displays:
  - Beam-penetration technique
  - Shadow mask technique

Beam-Penetration Techniques:

- This technique is used with random-scan monitors.
- In this technique, the inside of CRT screen is coated with two layers ofphosphor, usually red and green.
- The displayed color depends on how far the electron beam penetratesinto the phosphor layers.
- The outer layer is of red phosphor and the inner layer is of greenphosphor.
- A beam of slow electrons excites only the outer red color.
- A beam of very fast electrons penetrates through the red layer andexcites the inner green layer.
- At intermediate beam speeds, combination of red and green light areemitted and two additional colors, orange and yellow are displayed.
- The beam acceleration voltage controls the speed of electrons and hence the screen color at any point on the screen.

Merits:

- It is inexpensive technique to produce color in random scan monitors.

Demerits:

- It can display only four colors.
- The quality of picture produced by this technique is not good as compared to other techniques.

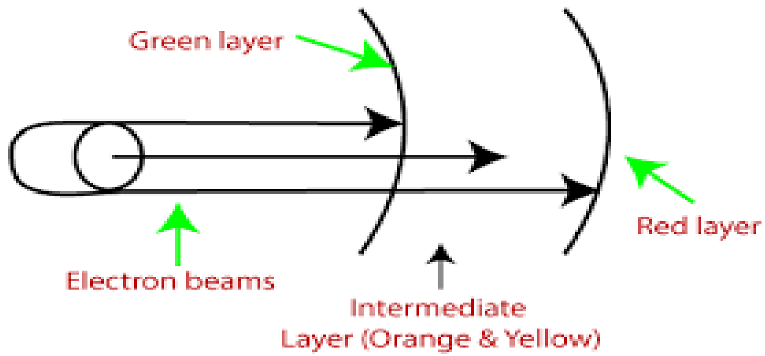


Fig 24. Beam Penetration Technique

Shadow Mask Technique:

- The shadow mask technique produces a much wider range of colorsthen the beam penetration technique.
- Hence, this technique is commonly used in raster-scan displays including color TV.
- In a shadow mask technique, CRT has three phosphor color dots ateach pixel position.
- One phosphor dot emits a red light, second emits green light and thirdemits blue light.
- It has three electron guns, one for each color dot and a shadow mask grid just behind the phosphor-coated screen.
- The shadow mask grid consists of series of holes aligned with the phosphor dot pattern.

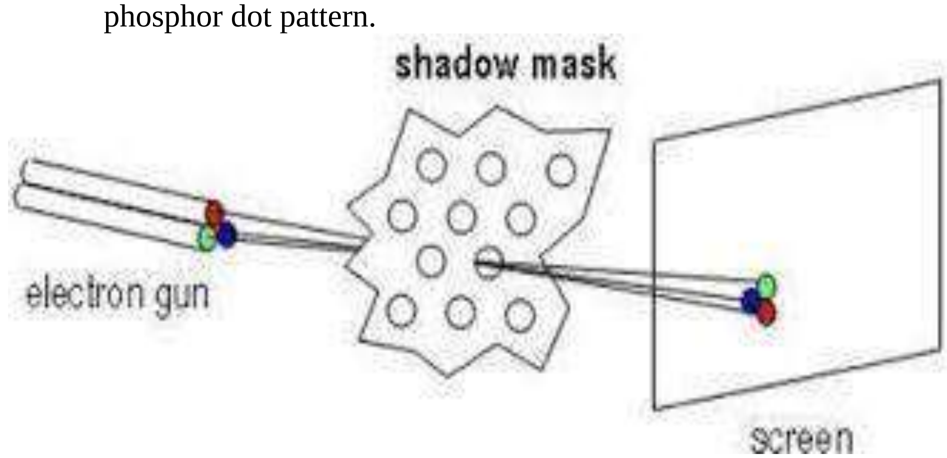


Fig 25. Shadow Mask Technique

Liquid Crystal Displays (LCD):

- The term liquid crystal refers to the fact that these compounds have acrystalline arrangement of molecules, yet they flow like a liquid.
- Two glass plates, each containing a light polarizer at the right angles tothe other plate squeezes the liquid-crystal material.
- Rows of the horizontal transparent conductors are built into one glassplate and columns of vertical conductors are put into the other

plate.

- The intersection of two conductors defines a pixel position.
- In the ON state, polarized light passing through material is twisted sothat it will pass through the opposite polarizer.
- It is then reflected back to the viewer.
- To turn OFF the pixel, we apply a voltage to the two intersectingconductors to align the molecules so that light is not twisted.

Merits:

- Less bulky.
- Suitable component for portable devices like laptops, mobile phones,etc.
- Power consumption is much less so ideal to be operated by batteries.

Demerits:

- More expensive than the CRT and more difficult to manufacture.

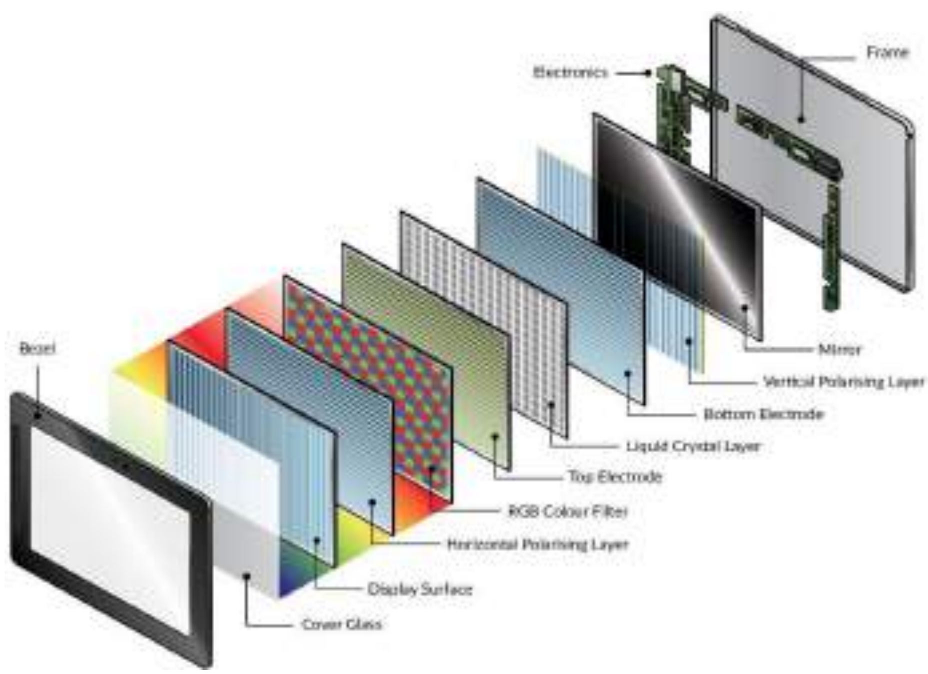


Fig 26. Working of LCD

LCD Displays V/S CRT Monitors:

LCD Displays	CRT Monitors
Thin, compact, lightweight. Takes up less space.	Bulky, heavier and takes up more space.
Lack in brightness than CRT.	Appreciable amount of brightness.

Consumes less electricity and produce less heat.	Consumes more power and produce more heat.
Much expensive.	Less expensive.
Portable.	Not portable.
Restricted viewing angles. It needs to be viewed from front to have better view.	Viewable from almost every angle.

RASTER SCAN SYSTEMS

- Hardware architecture of the raster scan system is shown in the Fig 9.
- In addition to CPU, a special-purpose processor, called video controller or display controller, is used to control the operation of the display device.
- In this architecture, frame buffer is allocated a dedicated area in system memory.
- Frame buffer can be anywhere in the system memory.

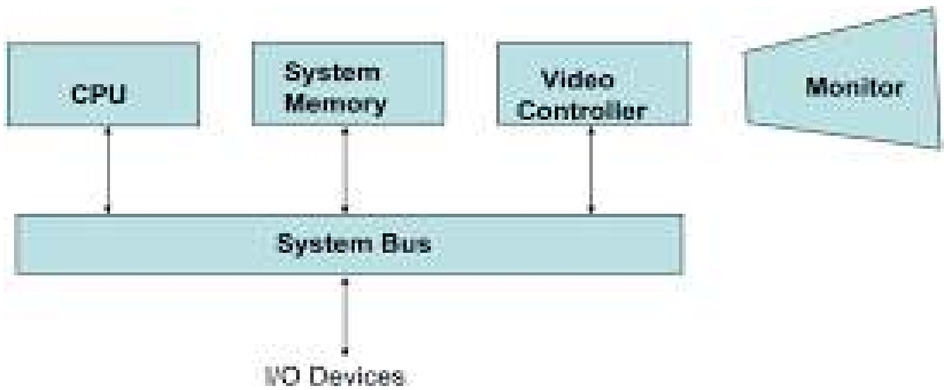


Fig 27. Architecture of a Raster Scan System

Video Controller:

- Another variant of a raster scan system is shown in Fig 10.
- A dedicated memory area is allocated to the frame buffer.
- Video controller is given direct access to frame buffer.
- The organization can render scene quickly because video controller do not have to put a request on system bus to read intensity value from frame buffer.
- In some system, origin is set at the bottom left location of the screen.

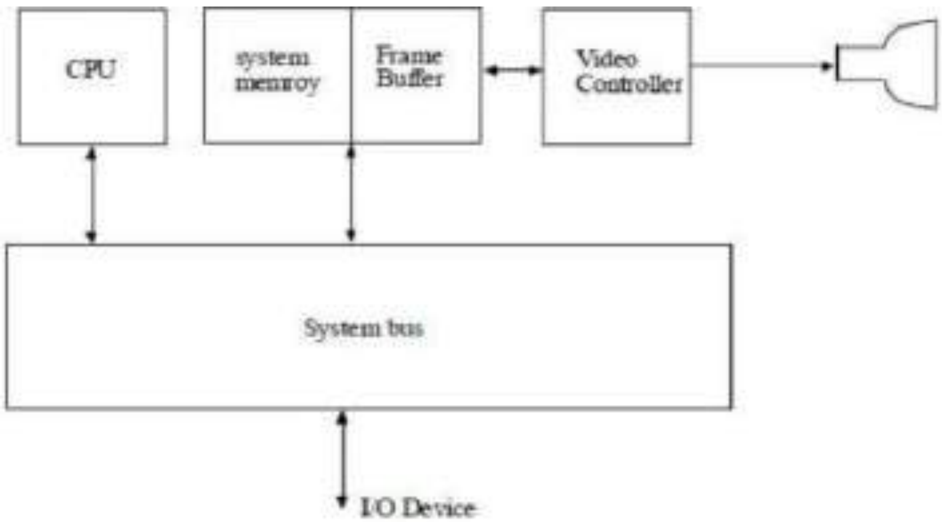


Fig 28. Architecture of a Raster Scan System

Display Processor:

- Display processor is also known as graphics controller or a displaycoprocessor.
- Its purpose is to free the CPU from graphics operations.
- The major task of a display processor is to digitize a picture definition given in an application program into a set of pixel intensity values for storage in frame buffer.
- This digitize process is called scan-conversion.
- Graphics commands specify straight lines and other geometric objectsare scan converted into a set of discrete intensity points.
- Scan converting a straight line means, we have to locate the pixel positions closest to the line path and store the intensity for each positionin the frame buffer.

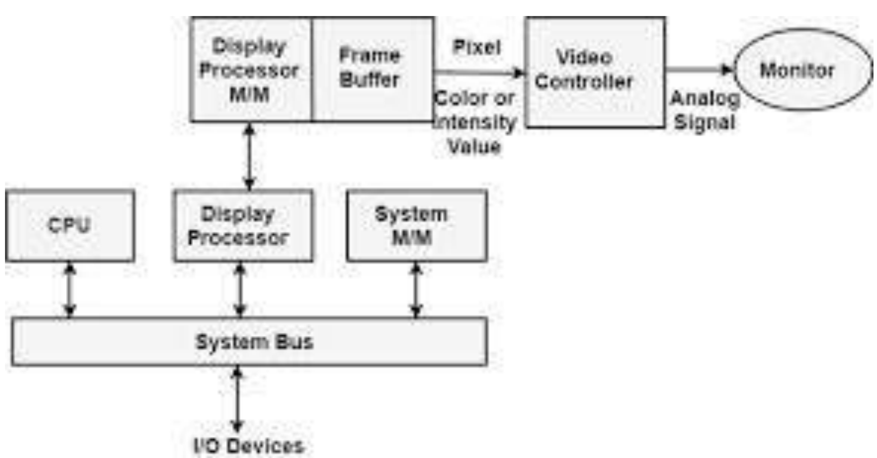


Fig 30. Raster Scan System with a display processor

RANDOM SCAN SYSTEMS

- Hardware architecture of random scan system is shown in Fig 13.

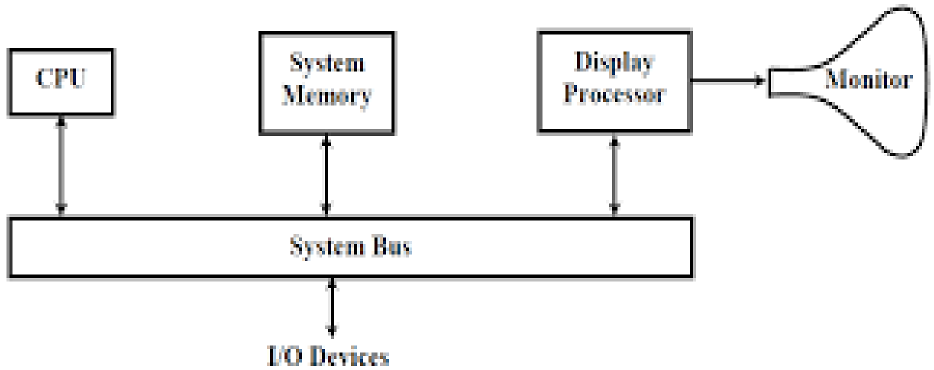


Fig 31. Random Scan System

- Application program resides in system memory.
- Graphics packages translate the graphics commands in the applicationprogram into a display file.
- Display file is also stored in system memory.
- Display processor accesses the content of display file and display onmonitor screen.
- Display processor retrieves one by one command from display file anddraw it on the screen.
- Display processor in random scan system is also called a graphics controller or display processing unit.
- Random scan devices render the scene using short line.
- Electron beam track the line position directly, rather than going througheach and every pixel on screen.
- Entire scene is defined using segment of line in in application program.

Module II: Scan conversion and 2D Graphics
2.1 Scan converting a Point.
2.2 Line (Direct, DDA and Bresenham line algorithms).
2.3 Circle (Direct, Polar, Bresenham and Mid-point circle algorithms).
2.4 Ellipse (Direct, Polar and Midpoint ellipse algorithms).
2.5 Area filling techniques (Boundary fill, Flood fill, scan line area fill algorithm).
2.6 Limitations of scan conversion.
2.7 2D Cartesian and Homogeneous co-ordinate system,
2.8 Geometric transformations (Translation, Scaling, Rotation, Reflection and Shearing).
2.9 Composite transformations.
2.10 2D dimensional viewing transformation and clipping (Cohen –Sutherland, Liang-Barsky, Sutherland-Hodge man algorithms).

INTRODUCTION OF SCAN CONVERSION

Scan conversion:

- Scan conversion is defined as a process of representing continuous graphic object as a collection of discrete pixels. It is also called as **Rasterization**.
- The graphics objects can be a point, line, circle, arc, rectangle, polygonand other shapes.
- Graphics objects are continuous.
- Pixels are always discrete. Each pixel has two states: on and off. 0 isaddressed by pixel off and 1 is addressed by pixel on.
- The video display device of the computer has a circuit that converts thebinary values of 0 and 1 into the information of pixel on and off.
- Image is represented as collection of pixels.

Pixel:

- Pixel is a smallest unit of a digital image or graphics object, which can be displayed and represented on a display device. A pixel is also called as a pel or picture element.
- A pixel is a basic logical unit in the graphics and the number of pixel are combined to form a complete image, video or text.
- A pixel is represented by a dot or a square on a display screen.
- Pixels are the basic building blocks of an image that are generated using geometric coordinates.

- Depending on the display monitor and graphics card, the size and color combination of pixel varies and is measured in terms of display resolution.

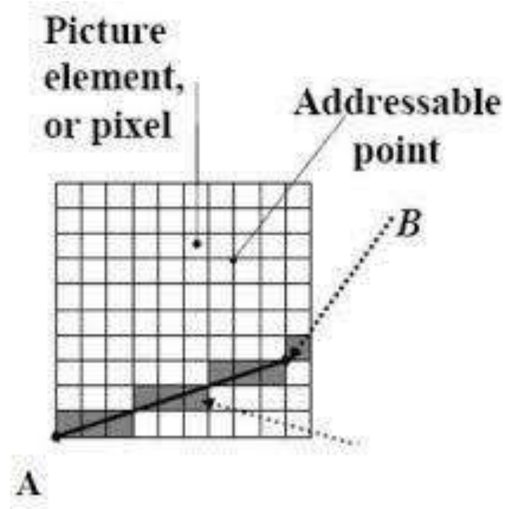


Fig 32. Pixel representation

SCAN CONVERSION OF POINTS

- Vectors are defined mathematically in an infinite, real number Cartesian Co-ordinate system.
- The physical coordinates of the pixel on the computer screen based upon the screen resolution is called as screen coordinates, device coordinates, or pixel coordinates.
- A display device deals with finite, discrete coordinates.
- Both x-coordinate and y-coordinate are positive values.
- On a display device, the origin (0, 0) is always on the top-left corner of the screen.

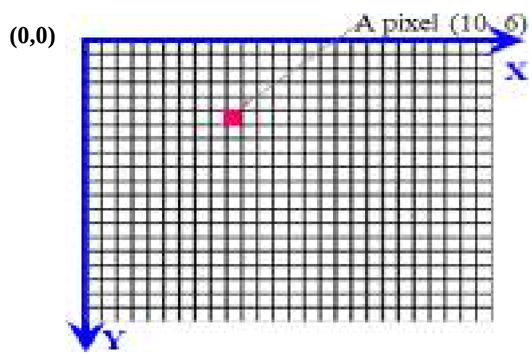


Fig 33. A pixel representation as screen coordinates on a displaydevice

- The values in the x-coordinate increases from left to right and thevalues of the y-coordinate increases from top to bottom.

- A mathematical point (x, y) where x and y are real numbers within animage area, needs to be scan-converted to a pixel location at (x’, y’).
- Each pixel on graphic display does not represent a mathematical pointlike P (2.9, 4.22).
- This may be done by making x’ and y’ to be the integer part of x and y,by applying any mathematical function.

Example:
Floor is a function that returns largest integer that is less than or equalto the argument

x’=Floor(x)
y’=Floor(y)
for point P (2.9, 4.22).
x’=Floor (2.9) and y’=Floor (4.22),
(x’, y’) will be (3, 4).

SCAN CONVERSION OF LINES

A line has two end points and is defined by an equation y=mx + c, where m is the slope of the line and c is the y intercept.

Consider the two end points as P1 (x1, y1) and P2 (x2, y2). The equation of the line is used to determine the x and y coordinates of all the points that lie between P1 and P2.

The line is drawn on the screen when there are two end points as P1 and P2, and the computer fills it with pixels by the intermediate point values. This can be done with various line drawing algorithms like DDA and Bresenham’s.

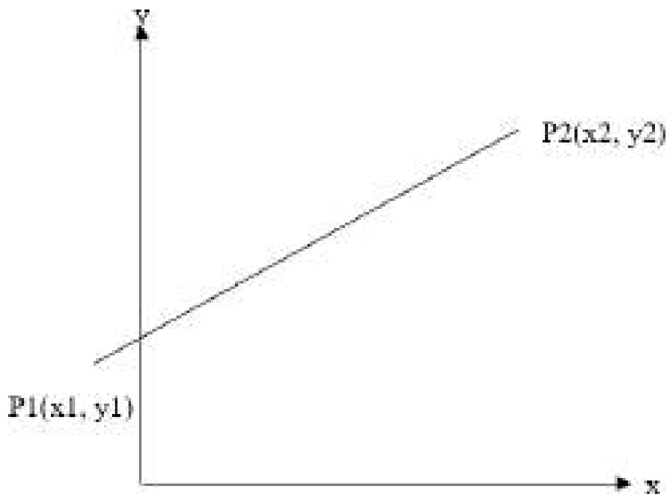


Fig 34. Line representation on coordinate system

DDA Line Drawing Algorithm

DDA stands for Digital Differential Analyzer. This algorithm works onincremental approach.

<u>Drawing Concept</u>

- The next coordinate to be plotted is calculated based on the previous coordinates.

- The line to be drawn is based on the analysis of difference between the pixel points.
- The incremental points are calculated based upon the start point and theend point of the line.
- It calculates pixel positions along a line by taking unit step increment along one direction and calculating corresponding coordinate position based on the rate of change of the coordinate (Δx or Δy).
- The slope of the line is the ratio of difference of y coordinates and difference of x coordinates.

$\Delta y = (y2 - y1)$

$\Delta x = (x2 - x1)$

where,

(x1, y1) is the start point

(x2, y2) is the end point

Δy is the difference between y coordinates

Δx is the difference between x coordinates

- The value of the slope will be either positive or negative.
- If the value of the slope is positive, then Δx and Δy are incremented,else decremented.
- For each pixel we move right (along the x-axis), we need to movedown (along the y-axis) by *m* pixels (based on screen coordinates).

Algorithm:

1. Start
2. Input the two end points of a line as (x1, y1) and (x2, y2)
3. Compute

   dx=x2-x1 and dy=y2-y1
4. if abs(dx) > abs(dy)

   step = abs(dx)

   else

   step = abs(dy)
5. Compute

   xinc = dx/step

   yinc = dy/step
6. Initialize x = x1 and y = y1
7. Plot pixel (x, y)
8. For k=1 to step

   {

```
        x = x + xincy
        = y + yinc
        Plot pixel (ROUND(x), ROUND(y));
    }
```

9. End

Merits:

- It is simple.
- It is fast to implement.

Demerits:

- It involves floating values, which requires rounding off to plot on screen.
- Rounding to integers takes time.

Example of DDA Line Drawing Algorithm:

- Consider a line PQ with P = (0, 0) and Q = (7, 4). Apply the simpleDDA algorithm and calculate the pixel on this line.

Solution to trace line PQ.

1. Start
2. Inputx1=0, y1=0, x2=7, y=4
3. Compute

   dx=x2-x1=7-0=7

   dy=y2-y1=4-0=4
4. abs(dx)=7 and abs(dy)=4, so abs(x)>abs(y)

   so, step=abs(dx)=7
5. Compute

   xinc=dx/step=7/7=1

   yinc=dy/step=4/7=0.6
6. Initialize x=0, y=0
7. Plot the pixel (0, 0)
8. For k=1 to 7,

   Calculate new values of x and y as x = x + xinc and y = y + yinc.

   Calculated values shown in Table 1.
9. End

Table 1. Pixel value to compute line PQ

k	Plot	x	y
	0, 0	0	0
1	1, 1	1	0.6
2	2, 1	2	1.2
3	3, 2	3	1.8
4	4, 2	4	2.4
5	5, 3	5	3
6	6, 4	6	3.6
7	7,4	7	4.2

Bresenham’s Line Drawing Algorithm:

Bresenham’s line drawing algorithm determines the points of an n- dimensional raster that should be selected in order to form a close approximation to a straight line between two points.

This algorithm provides the fast and efficient way to represent continuous line on a discrete plane of a computer display.

The algorithm is suitable for digital plotter and CRT.

Algorithm:

1. Start
2. Input the line end points (x1, y1) and (x2, y2)
3. Plot pixel (x1, y1)
4. Compute

   dx = x2 – x1

   dy = y2 – y1
5. Initialize p_k = 2*dy – dx
6. For each x along the line, starting at k = 0, perform the following test

   If p_k< 0,

   Plot pixel (x_k+1, y_k)

   p_k+1= p_k + 2*dy

   Else

   Plot pixel (x_k+1, y_k+1) p_k+1=

   p_k + 2*dy –2*dx
7. Repeat step 6, dx – 1 times.
8. Plot (x2, y2)
9. End

Merits:

- It generates only integer values, so simple to use.
- It is faster than DDA line drawing algorithm.
- It avoids generation of duplicate points.
- Points generated are more accurate.

Demerits:

- The resultant line obtained is not smooth.
- This algorithm is for basic line drawing. It cannot handle zigzag lines.

Example of Bresenham's Line Drawing Algorithm:

- Consider a line AB with A = (0, 0) and B = (7, 4). Apply the Bresenham's line algorithm and calculate the pixel on this line.
- Solution to trace line AB.

1. Start
2. Input x1=0, y1=0, x2=7, y=43.

   Plot (0, 0)

4. Compute

   dx=x2-x1=7-0=7

dy=y2-y1=4-0=4

5. Initialize p_k = 2*dy-dx,

    p_k = 2*4-7=8-7=1

6. For k=0 to 6,

    Calculate new values of x and y based upon decision parameterpk

    If p_k< 0,

    increment x, y will be same, plot (x_k+1, y_k)

    compute p_k+1 =p_k + 2*dy

    else

    increment x and y, plot (x_k+1, y_k+1)

    computep_k + 1 = pk+ 2*dy – 2*dx

    Calculated values shown in Table 2.

7. Repeat the steps dx-1=7-1=6 times

8. Plot (7, 4)

9. End

Table 2. Pixel value to compute line AB

k	**p_k**	**p_k+1**	**Plot**	**x**	**y**
			0, 0	0	0
0	1	-5	1, 1	1	0.6
1	-5	3	2, 1	2	1.2
2	3	-3	3, 2	3	1.8
3	-3	5	4, 2	4	2.4
4	5	-1	5, 3	5	3
5	-1	7	6, 4	6	3.6
6	7	1	7,4	7	4.2

SCAN CONVERSION OF CIRCLE

A circle is defined as a set of points that are at the same distance from center and the distance from center to any point on the circumference of the circle is called a radius.

- It is an eight-way symmetry as in Fig 6, which can be divided into four quadrants, and each quadrant into two octants.
- This symmetry helps in implementation of any circle-drawing algorithm.
- The symmetric property of the circle is utilized to scan convert the circle by plotting eight points for eight octants.

- This means for every point, eight points can be plotted as (x, y), (y, x), (-y, x), (-x, y), (-x, -y), (-y, -x), (y, -x) and (x, -y) as shown in Fig 6.
- The entire circle can be obtained by just computing the first octant and then repeat the reflection of the point about each 45^o axis.
- It produces seven more points on the circumference of the circle. Hence, it is sufficient to calculate only one octant.
- The circle centered at the origin can be described by the equation:$x^2 + y^2 = r^2$

where,

x and y are the coordinatesr

is the radius

- The second approach describe the circle using trigonometric relation, where x and y coordinates of the circle is given by x=r cosθ and y =r sinθ, and θ is the current angle.

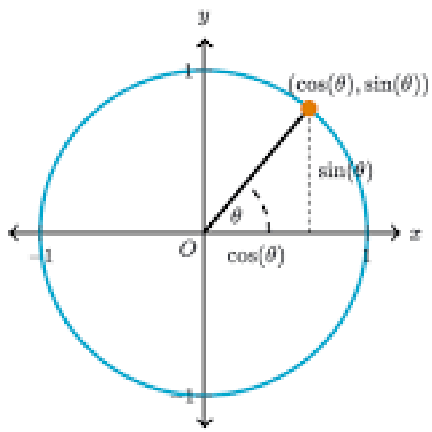


Fig 35. Representing circle in trigonometric relations

Midpoint Circle Algorithm:

This algorithm is used to determine the points needed for rasterizingthe circle.

- The perimeter points will be calculated for first octant and will beprinted along the mirror points in other octants.
- For any given point (x, y), the next point to be plotted can be (x, y+1)or (x-1, y-1) based upon the three conditions.
- Given a circle centered at (0, 0), with radius r and point P(x, y), $f(P)=x^2 + y^2 - r^2$
  - If f(P) < 0, the given point lies inside the circle boundary, then the upper pixel can be chosen.
  - If f(P) = 0, the given point lies on the circle boundary, then any pixel can be chosen.
  - If f(P) > 0, the given point lies outside the circle boundary, then the lower pixel is chosen.

Algorithm:

1. Start
2. Declare x, y, r, P, x_c, y_c where (x_c, y_c) are center coordinates, x and y points to be plotted, r is the radius, P is decision factor.
3. Initialize x = 0 and y = r.
4. Compute decision factor P = (5/4) – r.
5. Repeat the steps while $x \leq y$
6. Plot (x, y)
7. If P < 0

    Set P = P + 2x +3

    Else if $P \geq 0$

    Set P = P + 2(x-y) + 5

    y = y – 1
8. do x = x + 1
9. End

Merits:

- It is an efficient algorithm.
- It is easy to implement.
- It is based on simple circle equation $x^2 + y^2 = r^2$
- It helps to create curves on raster display.

Demerits:

- It is a time consuming algorithm.
- Sometimes the point of the circle are not accurate.

Bresenham's Circle Algorithm :

- The Bresenham's circle drawing algorithm calculates all the pointsnearest to the circle boundary.
- It is an incremental method where one coordinate of the point is incremented and another coordinate is calculated according to it.
- It uses integer arithmetic, which makes it less complex and working isfaster.
- The strategy is to select the pixel, which is at the least distance from the true circle boundary, and then keep calculating the successive points on the circle.
- The circle follows the symmetric property.

- In this algorithm, at any point (x, y) we have two options either to choose the next pixel at the top i.e. (x+1, y) or at the bottom i.e. (x+1, y-1).
- This is done using decision parameter d where, If d > 0, (x+1, y-1) will be the next pixel to plot Else (x+1, y) will be the next pixel
- To draw a circle with center (x_c, y_c) and radius r, it will start from (0, r)and move in first quadrant till x=y with initial conditions:

d=3-2*r x=0

and y=r

Algorithm:

1. Start
2. Declare x, y, r, D, x_c, y_c where (x_c, y_c) are center coordinates, x and y points to be plotted, r is the radius, D is decision factor.
3. Calculate decision parameter D as: D = 3–(2 * r)
4. Initialize x = 0, y = r
5. Compute next pixels on circle based upon decision parameter

   While x ≤ y

   Plot (x, y) if

   D<0, then

   D = D + 4x + 6

   else

   D = D + 4(x - y) + 10

   y = y – 1

   end if

$x = x + 1$

6. End

Merits:

- It is a simple algorithm.
- It is easy to implement.
- It is based on simple circle equation $x^2 + y^2 = r^2$

Demerits:

- It is not suitable for high graphics and complex images.
- Accuracy issues while generating points.

Midpoint Ellipse Algorithm:

Ellipse is defined as the geometric figure, which is set of all points on a plane whose distance from two fixed points known as foci remainsconstant.

- It has two axis, namely major axis and minor axis, where the majoraxis is the longest diameter and minor axis is the shortest diameter.
- Ellipse has a four-way symmetry property, which means quadrants aresymmetric and not the octants.
- Midpoint ellipse algorithm plots the point of an ellipse on the firstquadrant by dividing the quadrant into two regions.
- Each point (x, y) is projected to other three quadrants (-x, y), (x, -y)and (-x, -y) as it uses four-way symmetry.

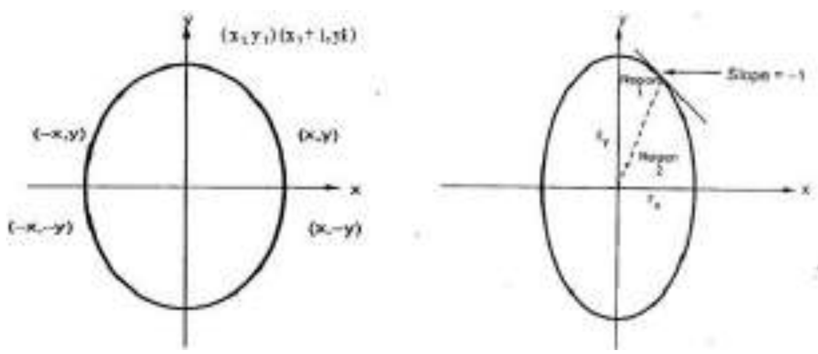


Fig 36. Four-way symmetry and Criteria for Ellipse

- Consider the equation of ellipse as $f(x, y) = r_y^2x^2 + r_x^2y^2 - r_x^2r_y^2$ where,

  r_x is the x radius

  r_y is the y radius
- For any point (x, y) the next point to be plotted is based on the threeconditions as:

- If $f(x, y) < 0$, then (x, y) is inside the ellipse
- If $f(x, y) > 0$, then (x, y) is outside the ellipse
- If $f(x, y) = 0$, then (x, y) is on the ellipse
- There will be two decision parameters based upon the two regions, P1for region 1 and P2 for region 2.
- For region 1, $P1 = r_y^2 + r_x^2 / 4 – r_y r_x^2$
- For region 2, $P2 = r_y^2(x + 1 / 2)^2 + r_x^2(y – 1)^2 – r_x^2 r_y^2$

ALIASING AND PROBLEMS IN ALIASING

- In computer graphics, smooth curves and other lines becomes jagged because the resolution of the graphics device or file is not high enough to represent smooth curve.
- In line drawing algorithms, not all rasterized locations match with the true line and we have to select the optimum raster locations to represent a straight line. This problem is more in low-resolution screens.
- In such screens, lines appear like a stair-step. This effect is called aliasing.
- The aliasing effect can be reduced by adjusting the intensities of the pixel along the line.
- This process of adjusting intensities of the pixels along the line to minimize the effect of aliasing is called antialiasing.
- The aliasing effect can be minimized by increasing resolution of the raster display device.

WHAT IS CLIPPING?

A procedure that identifies the portions of a picture that are either lying inside or outside of a specified region of space is referred to as clipping and can be done by using the clipping algorithms.

- Applications of clipping includes extracting part of a defined scene for viewing, identifying visible surface in three-dimensional views, object boundary, displaying a multi-window environment and many more. The clip window can be a general polygon or any curved boundary.
- For viewing transformation, we want to display only those picture partsthat are within the window area.
- Everything outside the window is discarded.
- Clipping algorithm can be applied in world coordinates, so that only the contents of the window interior are mapped to the device

coordinates.

- Clipping can be done on a point, lines, polygons and text.
- In Figure, lines and points within window should be kept (line (P5-P6) and point P8) and the line lying outside the window should be clipped (line (P1 to P2), line (P3 to P4) and point P7).

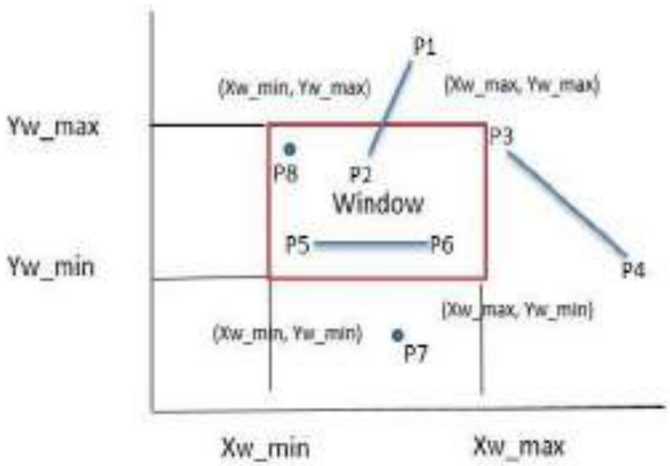


Fig 37. Example of clipping

POINT CLIPPING

Point clipping is a process to define the position of the point.

Point clip

Consider the clipping window is a rectangle in the standard position, apoint P (x, y) is used for display with the following constraints

$$xw_{min} \le x \le xw_{max} \text{ AND } yw_{min} \le y \le yw_{max}$$

where edges of the clip window are xw_{min}, xw_{max}, yw_{min} and yw_{max} then thepoint is clipped.

- In Figure, point B and point C are within the window range (xw_{min}, xw_{max}, yw_{min}, yw_{max}) so they are not clipped, while point A, point Dand point E lie outside the window so they are clipped.

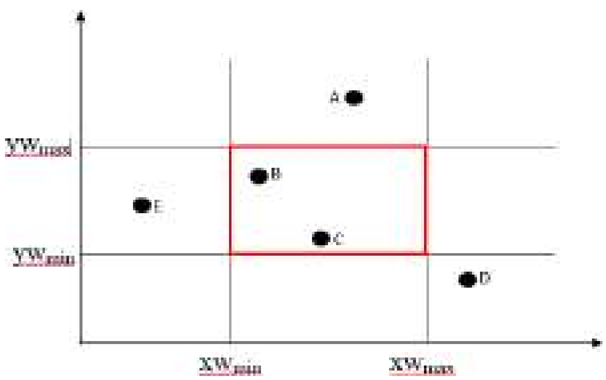


Fig 38. Example of Point Clipping

LINE CLIPPING

- Removing the portion of the lines which are lying outside the edges of the edges is called as line clipping.

Procedure for Line Clipping.

1. Test a given line segment to determine whether it lies completely inside the clipping window,
2. If it does not, we try to determine whether it lies completely outside the window.
3. If the line segment is not completely inside or completely outside, perform intersection calculations with one or more clipping boundaries, with the use of various line clipping algorithms.

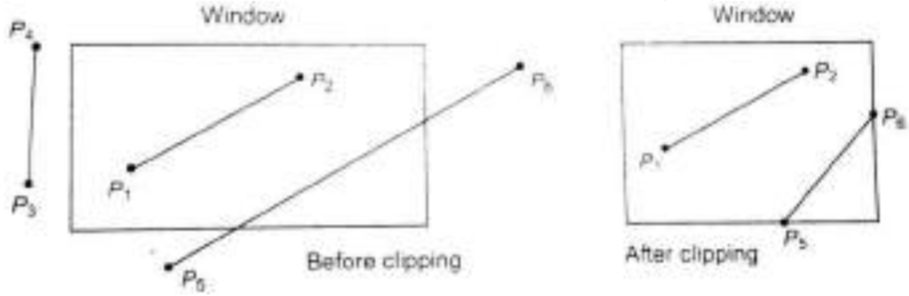


Fig 39. Example of Line Clipping

Examples

- In Figure, line P1 to P2 lies completely inside the window, so it does notrequires clipping and is saved.
- Line P3 to P4 lies completely outside the window, so it does not requires clipping and is not saved.
- Line P5 to P6 lies inside and outside partially, so it requires clippingand is clipped as shown in After Clipping of Fig 4.

Cyrus-Beck Line Clipping Algorithm:

Cyrus Beck Line clipping algorithm is a parametric line-clippingalgorithm. The term parametric means it is required to find the value of theparameter „t" in the parametric representation of the line segment for the point at that the segment intersects the clipping edge.

- In Figure, XY is the line segment, which is intersecting at the two edgesA and B of the convex window.
- Apply the parametric equation of the line segment

  XY as:X + t (Y-X)

where, $0 \leq t \leq 1$ and t is the linear parameter that continuously changesthe value

- Therefore,

X + t (Y-X) = (x1, y1) + t ((x2 – x1), (y2 – y1))

= (x, y) be any point on line XY

i. While t = 0, point X is obtained.
ii. While t = 1, point Y is obtained.
iii. While $0 \leq t \leq 1$, line in between point X and Y is traced.
iv. t = 0.5, finds the midpoint of line XY.
v. While t < 0, trace the line on left hand side of X.
vi. While t > 0, trace the line on right hand side of Y.

- The variation in parameter „t" generates line using pointwise method.
- The range of parameter values will identify the portion to be clipped through any convex polygonal region consisting of n-vertices or lattice points to be identified through the user.

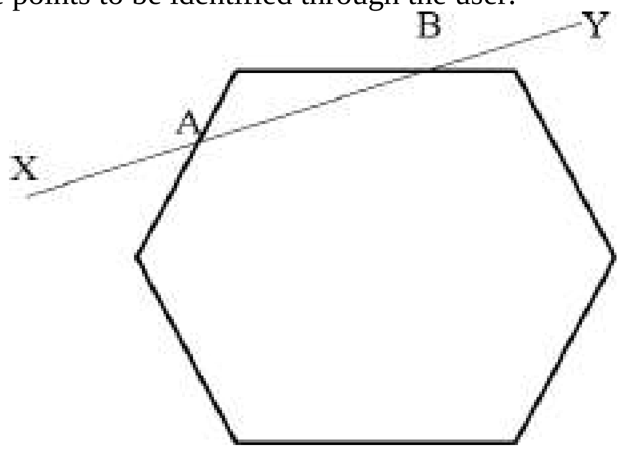


Fig 40. Interaction of line XY with convex polygon window

Merits:

- It follows parametric approach.
- It is fast and efficient.

Demerits:

- The algorithm is can be applicable only for 2D line clipping.

Cohen-Sutherland Line Clipping Algorithm :

Cohen-Sutherland algorithm is a line-clipping algorithm that cuts lines to portions, which are within a rectangular window.

- It removes the portion of the lines from a given set of lines and rectangle area of interest also known as view port, which belongs outside the area of interest and clip those lines that are partially inside the area of interest.

- The algorithm divides a two-dimensional space into 9 regions (one inside and eight outside) and then efficiently determine the lines and portions of lines that are visible in the central region of interest.
- Figure, defines the clipping regions for Cohen-Sutherland algorithm.

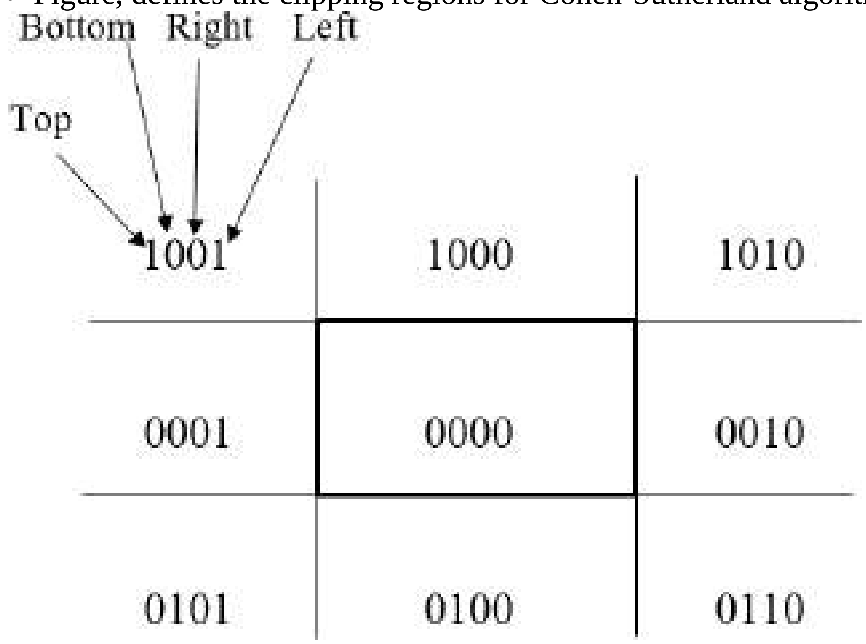


Fig 41. Clipping regions

- It uses 4-bits to divide the entire region, and the centre region is theclipping window.
- These 4 bits represents the TOP, BOTTOM, RIGHT and LEFT of theregion.
- In the Figure, the TOP and LEFT bit is set to 1 as it is in the TOP-LEFT corner.
- There are three possibilities for the line:
    - If the line is completely inside the window, the line is accepted.
    - If the line is completely outside the window, the line is rejected completely.
    - If the line is partially inside the window, find the intersection point and only draw that portion of the line that is inside the region.

Algorithm:

1. Assign a region code for each endpoint.
2. If both endpoints have a region code 0000 then accept this line.
3. Else perform the logical AND operation for both region codes

3.1 If the result is not 0000, then reject the line.

3.2 Else clipping is required.

 3.2.1 Choose an endpoint of the line that is outside the window.

 3.2.2 Find the intersection point at the window boundary based on the region code.

 3.2.3 Replace the endpoint with the intersection point and update the region code.

 3.2.4 Repeat step 2 until we find a clipped line either trivially accepted or trivially rejected.

4. Repeat step 1 for other lines.

Merits:

- This algorithm follows encoding approach.
- Easy to use

Demerits:

- This algorithm is applicable to rectangular window only and not to theother convex shaped window.
- Less efficient.

Liang-Barsky Line Clipping Algorithm :

This algorithm is more efficient than Cohen-Sutherland line clippingalgorithm and can be extended to 3-Dimensional clipping.

- It is a faster parametric line-clipping algorithm.
- It is based upon two concepts during clipping:
  - It is based upon the parametric equation of the line.
  - The intersections between the line and the clip window are determined by the inequalities describing the range of the clipping window.
  - The parametric equation is given

    by: x = x1 + t * dx

    y = y1 + t * dy

    where, dx = x2 – x1, dy = y2 – y1 and $0 \leq t \leq 1$
  - Liang-Barsky line-clipping algorithm uses four inequalities with two parameters p and q, as defined in the algorithm below.

Algorithm:

1. Start

2. Read the two endpoints of the line as p_1 (x1, y1) and p_2 (x2, y2).
3. Read the two corners of the clipping window as xw_{min}, yw_{min}, xw_{max}, yw_{max}.
4. Calculate the values of the parameters p_i and q_i for i = 1 to 4 such that:

   p_1 = -dx, $q_1 = x1 - xw_{min}$

   p_2 = dx, $q_2 = xw_{max} - x1$

   p_3 = -dy, $q_3 = y1 - yw_{min}$

   p_4 = dy, $q_4 = yw_{max} - y1$

5. If $p_i = 0$, then the line is parallel to i^{th} boundary.

   If $q_i < 0$, then:

   The line is completely outside the boundary so discard the line.

Else:

Check whether line is horizontal or vertical and then check theline

endpoints with the corresponding boundaries.

If the line endpoints lie within the bounded area then:Use

them to draw the line,

Else :

Use boundary coordinates to draw the line.

6. Initialize $t_1 = 0$ and $t_2 = 1$
7. Calculate q_i / p_i for i = 1 to 4
8. Select values of q_i / p_i where $p_i < 0$ and assign maximum out of them as t_1.
9. Select values of q_i / p_i where $p_i > 0$ and assign minimum out of them as t_2.
10. If (t1 < t2)

```
{
        newx1 = x1 + t1 * dxnewx2 = x1 + t2 * dxnewy1 = y1 + t1 * dynewy2 =
        y1 + t2 * dy
    Draw line (newx1, newx2, newy1, newy2)

}
```

11. End

Merits:

- It uses the parametric approach.
- It is more efficient than any other line-clipping algorithms.

POLYGON CLIPPING

A shape with set of connected lines is a polygon.

- Polygons are clipped based on the window, the portion inside the window is kept as it is and portion outside the window is clipped.
- It clips the four edges in the boundary of the clip rectangle.
- The clip boundary determines the visible and invisible regions of polygon clipping and it is categorized in four ways:
  - If visible region is fully inside the clip window, endpoints are saved.
  - If visible region is fully outside the clip window, no endpoints are saved.
  - If visible exits the clip window, save the intersection.
  - If the visible enters the clip window, save the endpoint and the intersection.
  - Each edge of the polygon must be tested against each edge of the clip rectangle, new edges must be added and existing edges must be discarded, retained or divided.

There are two types of polygons: (i) Concave (ii) Convex.

(i) Concave polygon:

It does not have any part of its diagonal in its exterior and at least one angle should be greater than 180^o.

(ii) Convex polygon:

It has at least one part of diagonal in its exterior and all angles should be less than 180^o.

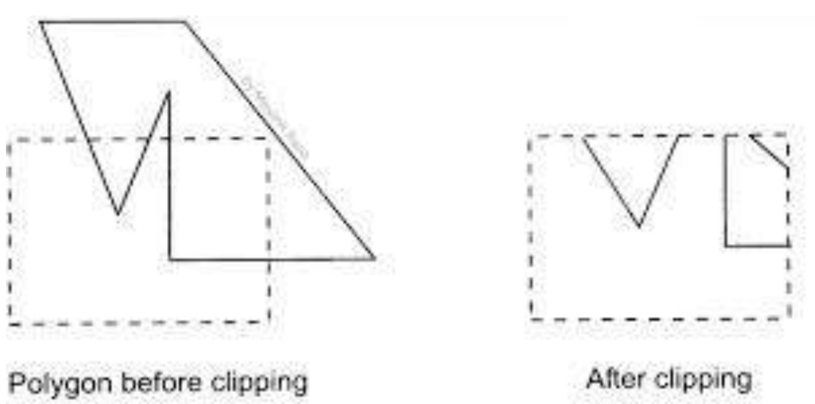


Fig 42. Polygon Clipping

Sutherland-Hodgeman Polgygon Clipping :

Sutherland-Hodgeman polygon-clipping algorithm uses divide and conquer strategy.

Procedure:

- It first clips the polygon against the right clipping boundary.
- The resulting polygon is then clipped against the top boundary.
- The same process is repeated for remaining two boundaries.
- It works in order and makes sure that all edges of clipping polygon are taken sequentially.
- At each step new sequence of vertices are generated and passed to next window boundary for clipping.
- To clip the vertices following cases are considered:

Case 1- Both vertices are inside:

- Only the second vertex is added to the output list.

***Case 2* -** First vertex is outside while second one is inside:

- Both the point of intersection of the edge with the clip boundary and the second vertex are added to the output list.

***Case 3* -** First vertex is inside while second one is outside:

- Only the point of intersection of the edge with the clip boundary is added to the output list.

***Case 4* -** Both vertices are outside:

- No vertices are added to the output list.

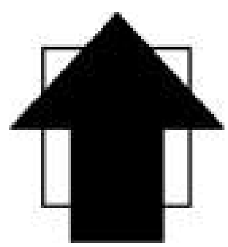


Fig 43. Before Clipping

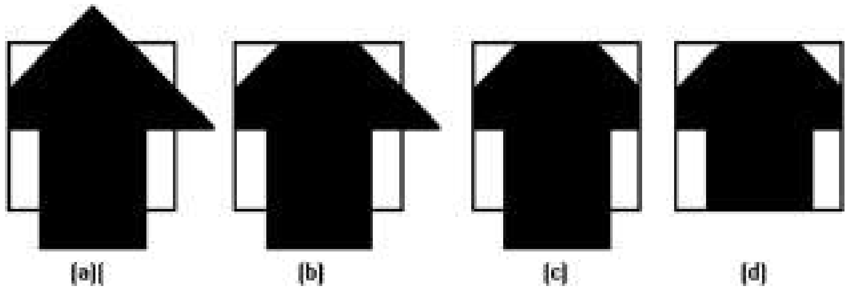


Fig 44. After clipping using Sutherland-Hodgeman algorithm

Example:

a. Clipping against the left side of the clip window.
b. Clipping against the top side of the clip window.
c. Clipping against the right side of the clip window.
d. Clipping against the bottom side of the clip window.

Sutherland-Hodgeman comprises of two key processes in the algorithm:

- Perform inside-outside test to determine the visibility of a point or vertex.
- Determine the intersection of the polygon edge and the clipping

plane.Method to determine the visibility of a vertex or a point.

- Let V be the vertex and AB the window boundary.
- Three points A, B and V define Vector AB and AV line in the plane.
- If this plane is considered in the xy plane, then the vector cross product AV x AB has a z component given by:

  $(x_v-x_A)(y_B-y_A)-(y_v-y_A)(x_B-x_A)$

  - The sign of the z-component decides the position of point V withrespect to window boundary.
  - It can be as follows:

- $z>0$: point is on the right side of window boundary
- $z=0$: point lies on the window edge
- $z<0$: point is on the left side of window boundary

Algorithm:

1. Read the coordinate of vertices of the subject polygon and clipping polygon.
2. Consider an edge of the clipping window, compare the vertices of each edge of the subject polygon with the clipping polygon or the edge, and record the intersections.
3. Store the new intersection and vertices in the new list of the vertices as per the cases.
4. Perform steps 2 and 3 for the remaining edges of the clipping polygon

each time, the resulting list of polygon vertices are successively passed to process the next edge of the clipping polygon.

5. Finish.

Merits:

- The process is simple and efficient.

Demerits:

This method requires a considerable amount of memory.

TRANSFORMATIONS AND MATRICES

Transformation:

Creation of graphs and images in computer are not sufficient in computer graphics. In real world object are moved from one position to another or its changing its shape or orientation and it can be seen from different angles and different perspective.

The process of changing size, shape, position and orientation of object is called transformation.

Matrices:

- It provides the technique to represent the real world object into computer system.
- It also provides a way to calculate different type of geometric functionon the real life object to show on the screen.
- With help of matrices various operation like translation scaling etc can be done on the object

TRANSFORMATION CONVENTIONS

To draw an object onto screen, computer need the system that is called as conventions of transformation. This can be done with 2 types such as 2D coordinate system and 3D coordinate system.

For 2D image we use 2D coordinate system as convention for transformation whereas for 3D image we use 3D coordinate system as convention for transformation of image.

2D COORDINATESYSTEM:

This 2D coordinate system consists of two coordinates x and y i.e. x planeand y plane. Its direction is along with the x and y axis with reference to an origin.

With the help of this system the real word object can be draw onscreen with lines, circle, polygon etc

3D coordinate system:

This 3D coordinate system consists of three coordinates x, y and z .i.e. x plane, y plane and z plane.

Its direction is along with the x, y and z axis with reference to an origin. With the help of this system the real word object can be drawn on screen with lines, circle, polygon etc.

2D TRANSFORMATIONS

The transformation which is applied on 2 plane i.e. x and y plane to change the appearance of the object is called 2D transformation.

Basic 2D transformation	1. Translation	Denoted by T
	2. Rotation	Denoted by R
	3. Scaling	Denoted by S
	4. Shearing	Denoted by Sh
	5. Reflection	Denoted by RE

Translation:

- Changing the position of an object or shifting of an object is called 2Dtranslation.
- To achieve translation an object should shift in x and y direction.
- We can translate a two dimensional figure by adding **translation factor t_x and t_y** to the original points x and y

Rotation:

Changing the position of an object with rotation angle along with xyplane is called 2D rotation.

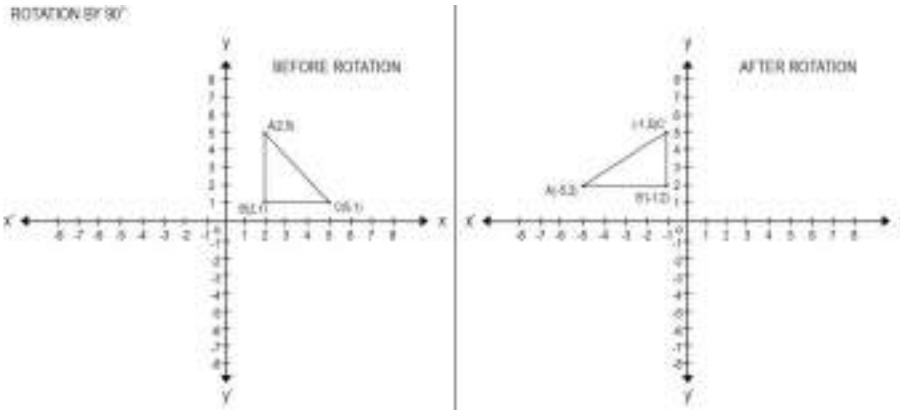


Fig 45. Rotation

Shearing:

- A shear is something that pushes the object sideway and make a thinglittle bit tilled.
- To achieve shearing an object should tilled(slant) with respect to x andy direction.
- We can shear a two dimensional figure by multiplying **shearing factorsh$_x$ and sh$_y$** to the original points x and y

Reflection:

- Creating mirror image of an object is called 2D reflection.
- To achieve reflection an object should appear in the form of mirror object in x and y direction.
- We can reflect a two dimensional figure by multiplying **reflectionfactor RE$_x$ and RE$_y$** to the original points x and y

COMBINED TRANSFORMATION

Sometime an object may require more than one transformation to get desired output, we need to transform an object at several times with different transformation like rotation, scaling, translation, shearing,reflection such a group of transformation is called combined transformation.

The sequence of transformation is applied on an object to get the required shape of object.

It is also called composite transformation.

Method:

When we need to apply more than one transformation on an object first we need to do single transformation in matrix form the we need to apply second transformation on an object in matrix form then the final matrix multiplication is considered for the desired object transformation.

HOMOGENEOUS COORDINATES AND MATRIX REPRESENTATION OF 2D TRANSFORMATIONS

TRANSFORMATION OF POINTS

All types of transformation such as translation, rotation, scaling, reflection, shearing is applied to get the transformation of point. All the transformation elements can be described through matrices to performtransformation on points.

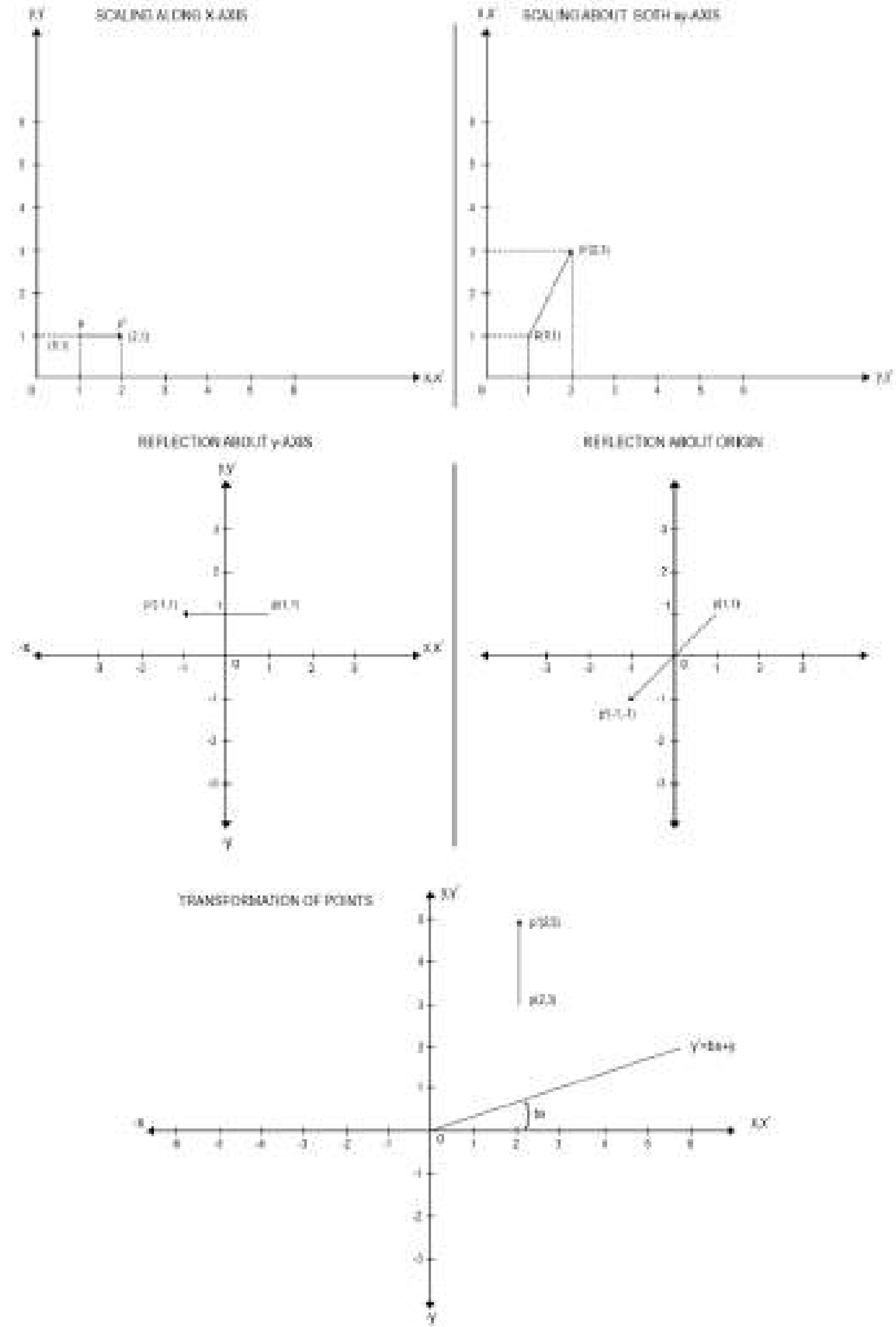


Fig 46 Transformation of point

TRANSFORMATION OF THE UNIT SQUARE

When we transform an object in 2D transformation, all the points are transformed except the origin point it will remain unchanged, which indicates that the conversion of an object in to new shape in 2D plane with xy coordinates. These points are considered as apposition vectors of a unit square with one point situated at the origin of the coordinate system. are same but the rest of the coordinates are changes according to the unit vector , hence the transformation is applied on all point except origin point.

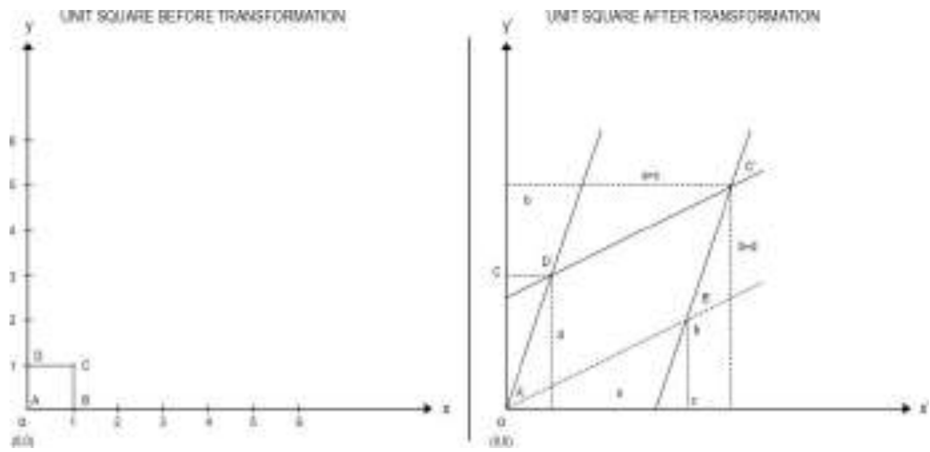


Fig 47 Unit square transformation

SOLID BODY TRANSFORMATIONS

- Solid body transformation is based on vector transformation. Before performing transformation basic physics quantities it should be known.
- We know that in solid body transformation parallel lines are transformed as parallel lines but it is not necessary that perpendicular line will always transformed perpendicular only.
- Angle between lines are also not preserved.
- To preserve an angle between two intersecting line solid body transformation is used. To achieve this, dot product and dot product ofvector are required.

Vector: A vector is quantity which has direction as well as magnitude. Itis used to determine the position of one point in space to another. Vector is used in graphics also known as vector graphics. Vector graphics is the creation of digital image through sequence of commands of mathematical statements that places lines and shapes in given two dimensional or three dimensional space. Vector graphics describes series of points to be connected.

Dot Product of vector: they can be multiplied using dot product of matrix is written using central dot present between these two vector.

Cross Product of vector: the cross product of two vector is an anothervector which is right angled to both the vector like a x b

ROTATION ABOUT AN ARBITRARY POINT

In translation and homogeneous coordinate system we have seen that the homogeneous coordinate system accomplishes the rotation about the points of an origin whereas, 2D transformation does not allow rotation of points at origin.

REFLECTION THROUGH AN ARBITRARY LINE

Reflection is achieved by mirroring the line coordinates on x axis, y axis and even by passing line through origin when reflection is taken place on both x and y axis.

We know that when x and y both the coordinates are changed, the line is reflected and passes through the origin. But sometimes reflection of an object through a line which does not pass through an origin id required.

Step 1: Translation: translate the line and the object so that the line passes through origin.[T]

Step 2: Rotation: Rotate the line and the object the origin until the line is coinciding with one of the coordinate axis. [R]

Step 3: Reflection: reflect through the coordinate axis [R]

Step 4: Inverse rotation: apply inverse rotation about the origin. $[R]^{-1}$

Step 5: Inverse translation: translate back to original location. $[T]^{-1}$

VIEWPORT TRANSFORMATIONS

A world coordinate area selected for display is called a **window**. An area on display device to which a window is mapped is called **viewport.** The window defines what to be display on screen whereas, the viewport defines where is to be displayed. The standard rectangle position is applied to see window and viewport**.** The part of real word image (world coordinate system) seen on device coordinate system (screen coordinate system) is called as **viewing transformation.** Sometimes two dimensional transformations is also called as **window to viewport transformation**.

In computer graphics, the term window is referred to an area of image that is to be display on the screen (View plane). Whenever we want to display image on screen the viewing coordinate is rectangular window which shows the enlarge image in viewport. By changing the position of the viewport, we can view object at different position on the display area of output device. Also, by varying the size of viewports, we can change the size and proportions of displayed objects. When all coordinate transformations are completed, viewport clipping can be performed in normalized coordinates or in device coordinates. Viewport is typically defines in unit square.

Process of window to viewport coordinate system:

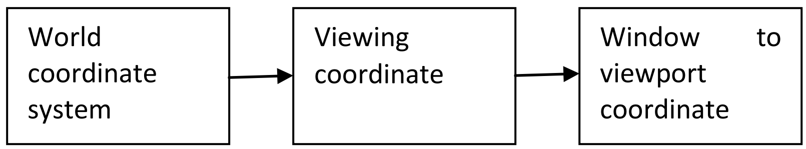


Fig 48 window to viewport system

Window to viewport mapping:

A point at position (xw, yw) in the window is mapped into new position (xv , yv) in associated view port.

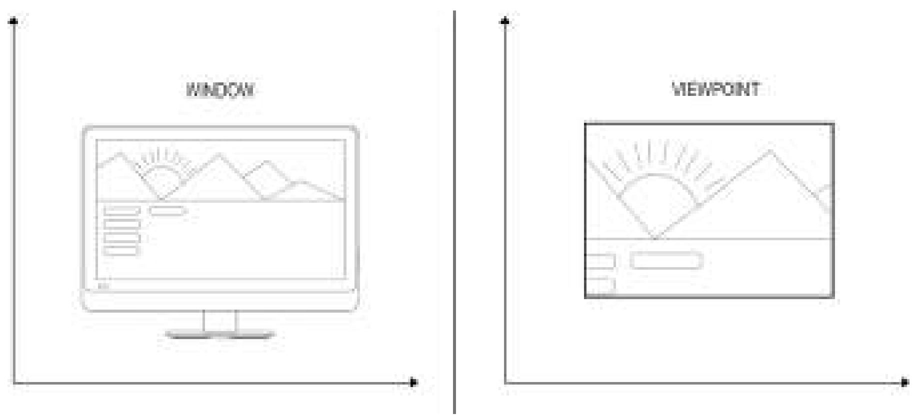


Fig 49 Window to Viewport transformation

Module III: 3D Graphics
3.1 3D Cartesian and Homogeneous co-ordinate system,
3.2 Geometric transformations (Translation, Scaling, Rotation, Reflection),
3.3 Composite transformations.
3.4 Mathematics of Projections: Perspective Projections - Mathematical Description and Anomalies of perspective projections.
3.5 Parallel Projections – Taxonomy of Parallel Projections and their Mathematical Description.
3.6 Introduction to 3D viewing pipeline and 3D clipping.

INTRODUCTION

When we want to convert the real life object into computer weneed to understand the coordinate system. The real life objects represent in computer with the different angles and different perspective with the help of different coordinate system and here the concept of transformation exist. Transformation takes place in x axis, y axis, z axis in 3D transformation. In this chapter we will see the 3D coordinate system of converting real life object to computer. The transformation which takes place on 3D plane or 3D space is called as 3D transformations.

TRANSFORMATIONS AND MATRICES

Transformation:

Creation of graphs and images in computer are not sufficient in computer graphics. In real world objects are moved from one position to another or its changing its shape or orientation and it can be seen from different angles and different perspective.

The process of changing size, shape, position and orientation of object iscalled transformation.

This change can be done by **Geometric transformation:**

In this type of transformation the object changes its appearance with thehelp Geometric properties of the specified points.1. Coordinate transformation:

In this type of transformation the object y' appears with the modified points on the screen after performing some non-singular x' operation like translation, rotation, scaling etc.

Matrices:

- In computer graphics, objects can be described in various quantitative numbers also called points. To represent these quantities, matrices are used.
- It provides the technique to represent the real world object into computer system.
- It also provides a way to calculate different type of geometric function on the real life object to show on the screen.

- With help of matrices various operation like translation scaling etc can be done on the object
- General Matrix representation of 3 × 3 matrix $\begin{bmatrix} 1 & 4 & 3 \\ 5 & 9 & 8 \\ 8 & 2 & 4 \end{bmatrix}$
- Homogeneous Matrix representation of 3 × 3 matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

TRANSFORMATION CONVENTIONS

To draw an object onto screen, computer need the system that is called as conventions of transformation. This can be done with 2 types such as 2D coordinate system and 3D coordinate system.

For 2D image we use 2D coordinate system as convention for transformation whereas for 3D image we use 3D coordinate system as convention for transformation of image.

3D coordinate system:

- This system is made up of three coordinates: x, y, and z, which correspond to the x, y, and z planes, respectively and its direction along with the x, y and z axis with reference to an origin. With the help of this system, the real word object can be drawn on screen with lines, circles, polygons, etc. Rules applied for transformation

Right hand coordinate system:

- In this coordinate system right hand figures (X-Thumb, Y-index and X-middle) used as axis to represent the position of the transformed image.
- In Right hand coordinate system all coordinates are positive so the object will transform x in positive direction only.

Left hand coordinate system:

- In this coordinate system right hand figures(X-Thumb, Y-index and Z-middle) used as axis to represent the position of the transformed image.
- In Right hand coordinate system x and y coordinates are positive and z coordinate is negative so the object will transform in both the direction.

3D TRANSFORMATIONS

The transformation which is applied on 3 plane i.e. x, y and yz plane to change the appearance of the object is called 3D transformation.

Basic 3D transformation	1.	Translation	Denoted by T
	2.	Rotation	Denoted by R
	3.	Scaling	Denoted by S
	4.	Shearing	Denoted by Sh
	5.	Reflection	Denoted by RE

Translation

- Changing the position of an object or shifting of an object is called 3D translation.
- To achieve translation, an object should shift in the x, y, and z direction.
- We can translate a two dimensional figure by adding **translation factor t_x, t_y and t_z** to the original points x, y and z.

Rotation:

- Changing the position of an object with rotation angle along with xyz plane is called 3D rotation.
- To achieve rotation an object should move in x and y direction with the specified angle □ .
- We can rotate a two dimensional figure by multiplying **rotation factor r_x , r_y and r_z** to the original points x, y and z

Scaling:

- Changing the size of an object is called 3D scaling.
- To achieve scaling an object should change the size of object in x, y and z direction.
- We can scale a two dimensional figure by multiplying **scaling factor s_x , s_y and s_z** to the original points x, y and z
- There are two types of scaling
  1 Uniform scaling: if all scaling factors are same then it is called as Uniform Scaling
  2 Non-uniform scaling: if all scaling factors are different then it is called as Non- Uniform Scaling

Shearing:

- A shear is something that pushes the object sideway and make a thing little bit tilled.
- To achieve shearing an object should tilled(slant) with respect to x, y and z direction.
- We can shear a two dimensional figure by multiplying **shearing factor sh_x , sh_y and sh_z** to the original points x, y and z.

Reflection:

- Creating mirror image of an object is called 3D reflection.
- To achieve reflection, an object should appear in the form of mirror object in x, y and z direction.
- We can reflect a two dimensional figure by multiplying **reflection factor RE_x , RE_y and RE_z** to the original points x, y and z

MULTIPLE TRANSFORMATION

Sometime an object may require more than one transformation to get desired output, we need to transform an object at several times with different transformation like rotation, scaling, translation, and shearing, reflection such a group of transformation is called combined transformation.

Multiple transformation means that applying more than one transformation on single object by using respective transformation matrix. The sequence of transformation is applied on an object to get the required shape of object.

Method:

When we need to apply more than one transformation on an object first we need to do single transformation in matrix form the we need to apply second transformation on an object in matrix form then the final matrix multiplication is considered for the desired object transformation.

HOMOGENEOUS COORDINATES AND MATRIX REPRESENTATION OF 3D TRANSFORMATIONS

In 2D coordinate system homogeneous coordinate can be represented in 3 X 3 matrix whereas, in 3D coordinate system homogeneous coordinate can be represented in 4 X 34 matrix.

Hence instead of only x, y, z in 3D coordinate system we could use x, y, z, h where $h \neq 0$

ROTATION ABOUT AN ARBITRARY AXIS INSPACE

Rotation about an arbitrary axis is implemented using translation and rotation about the coordinate axis. Hence there should be coordinate axis which is coincident to the arbitrary rotation axis. In 3D we have rotation about an arbitrary point through coordinate axis which is rotate around itself of the object

Steps to achieve Rotation about an arbitrary Axis in space

1. The origin point which is to be translate p=[a_x, b_y, c_z] trough all 3 axis

$$[T]=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -ax & -by & -cz & 1 \end{bmatrix}$$

2. Perform specified rotation about an arbitrary axis x and y. Rotation in anticlockwise angle implies positive direction.
3. Apply Inverse rotation on step 2
4. Inverse Translation of step 1 to make an arbitrary axis in a particular direction except towards origin.

REFLECTION THROUGH AN ARBITRARY PLANE

Reflection is achieved by mirroring the line coordinates on x axis, y axis and even by passing line through origin when reflection is taken place on both x and y axis.

We know that when x and y both the coordinates are changed; the line is reflected and passes through the origin. But sometimes reflection of an object through a line which does not pass through an origin id required.

Step 1: Translation: translate the point P which is lies on reflection plane. [T]

Step 2: Rotation: Rotate the point about x and y axis. [R]

Step 3: Reflection: reflect an object through z axis or xy plane with z=0[R□]

Step 4: Inverse rotation: apply inverse rotation about 2 axes which state instep 2. $[R]^{-1}$

Step 5: Inverse translation: perform inverse translation of step 1 to getback to original location. $[T]^{-1}$

AFFINE AND PERSPECTIVE GEOMETRY

- Study of parallel lines in an object is called affine geometry.
- In affine geometry lines of object are always parallel whereas in perspective geometry lines of object are always perpendicular means not parallel.
- Affine transformation can be achieved by combining translation, scaling, rotation and shearing together on object.
- In affine geometry study of connected lines are concerned.
- To convert one object shape into another shape in three dimensions

affine and perspective transformation is used.

- When 3D object is transformed into 2D object it is called as projection.
- In perspective projection the object is projected by center of projection of 3D image on 2d plane. If the center of projection is located at a finite point in 3D space then it is called as perspective projection.
- Another type of projection in which the center of projection has been move to infinite is called parallel projection.
- In parallel projection the information of the origin object remains same and the maintain realistic view of an image.

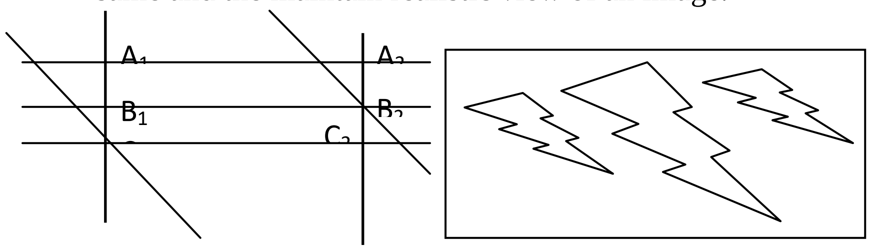


Fig 50 affine transformation and geometry

PERSPECTIVE TRANSFORMATIONS

When human eyes see far away things they are looking smaller as they come closer they are looking bigger to eyes this type of viewing is called perspective. This technique need to convert an object from one form to another and that is known as projection. Perspective projection will convert 3D image into 2D as human eyes does. The projection transforms shape of object into another form as they appear close to eyes.

There are three types of perspective transformation

1. Single point perspective transformation
2. Two point perspective transformation
3. Three point perspective transformation

1. Single point perspective transformation: If one term (either x, y, z) in fourth column of matrix transformation is non zero we will get single point perspective transformation.

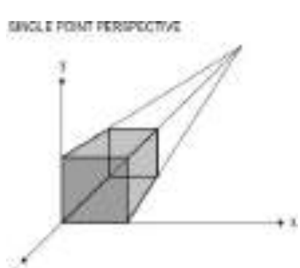


Fig 51 Single point perspective transformation

1. **Two point perspective transformation:** If two term (either x, y, z) in fourth column of matrix transformation is non zero we will get single point perspective transformation. In this type object can be seen in two directions.

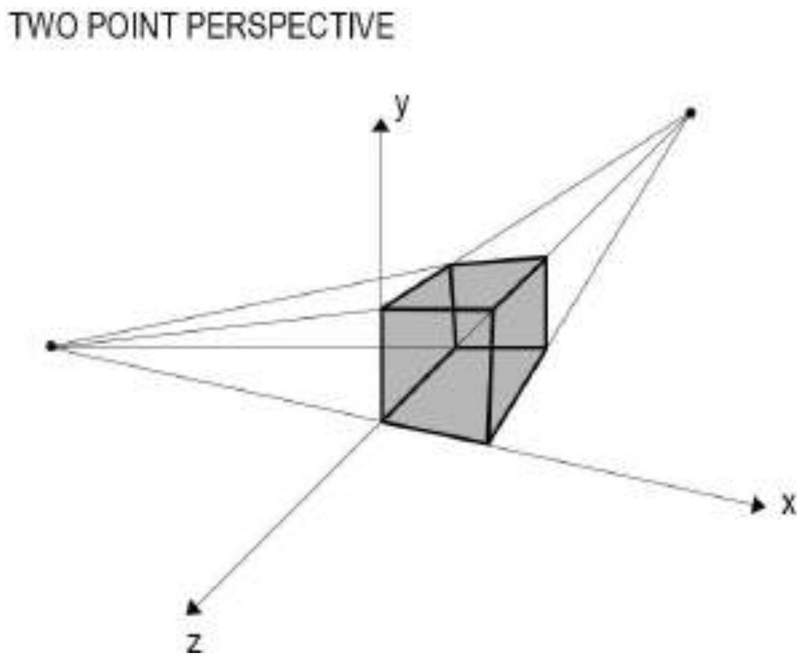


Fig 52 Two point perspective transformation

2. **Three point perspective transformation:** If three term (either x, y, z) in fourth column of matrix transformation is non zero we will get single point perspective transformation. In this type object can be seen in three directions.

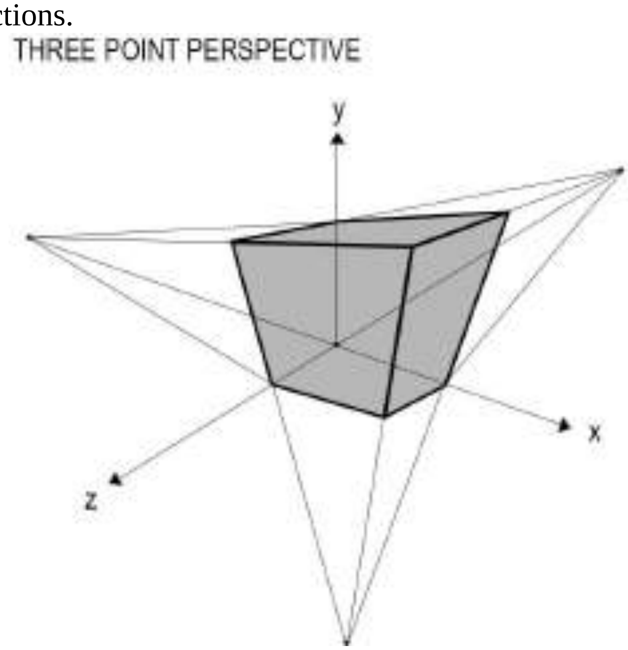


Fig 53 Three point perspective transformation

TECHNIQUES FOR GENERATING PERSPECTIVEVIEWS

Translation and rotation operations are used to create perspective viewof an object. When we see object from multiple directions, the object makes 3D view. For a fixed center of projection, a single point perspectiveprojection is carried out by multiple faces (One, two and threeperspective). Ref fig 6.10.1, 6.10.2, 6.10.3

1. **Translation of object**
2. **Rotation**

AXONOMETRIC PROJECTION

The problem of rendering for two-dimensional view of dimensional object existed long before the computer was used. One class of projection often used in drafting is called an **axonometric projection**. This is a parallel projection for which the direction of projection is perpendicular to the view plane. We can alter the direction of an **axonometric projection** to get different view of an object, provided we also change the view plane normal to match the projection direction. If we change our direction slightly to the side, then one of the side faces will become visible, while the edges of the front face will shorten. If we raise our angle of view top edges lengthen make the top face visible, while the edges on the side faces appear to shrink.

Types of axonometric projection are

1. **Isometric projection**
2. **Dimetric projection**
3. **Trimetric Projection**

1. Isometric projection: It is mainly used in engineering drawing. In this type of projection the object is viewing from all the direction is looking same and the principal axis are equally distributed on same distance with common angle of 120□ among the entire axis. Because of the equal angle among all three axis the distance and the length of the all side of object will be same. All The parallel lines are equal in length.

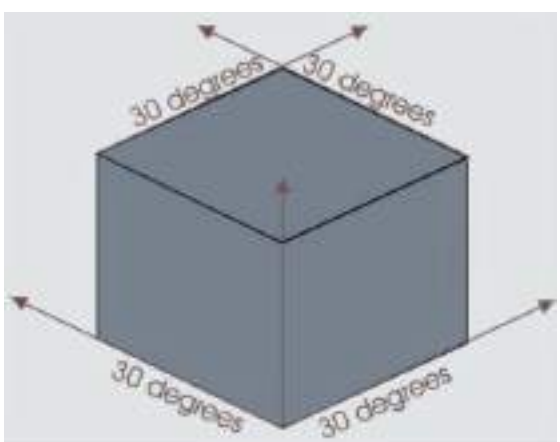


Fig 54 Isometric Projection

2. Dimetric projection: In isometric projection all three axis appears to be equally distanced but in Diemetric projection instead of all three, any two axis are equally distanced and one axis is different distanced than other two. This projection says that the view created through principal axis are represented the angle of object. Since two axis are same the line which are drawn along axis are also same.

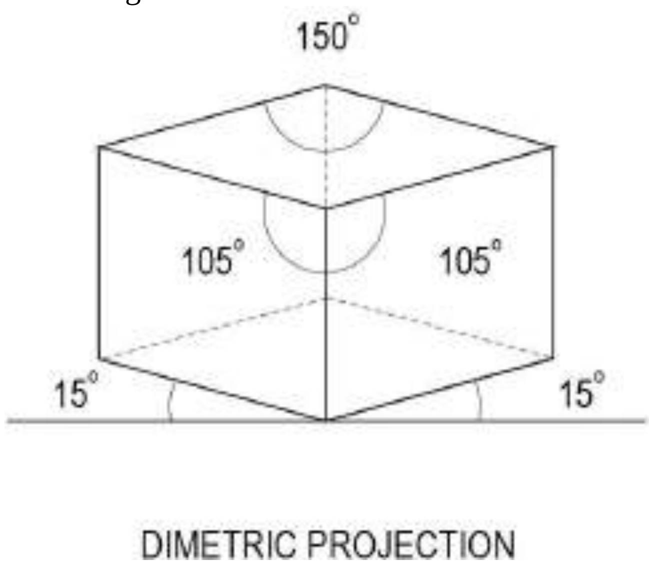


Fig 55 Dimetric Projection

3. Trimetric Projection: isometric projection all three axis appears to be not equally distanced. All three axis are unequally distanced as well as the angles created by these lines are also unequal.

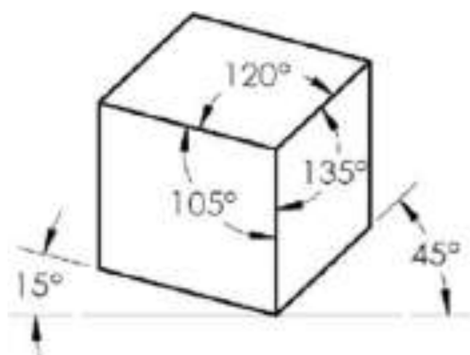


Fig 56 Trimetric Projection

OBLIQUE PROJECTION

If the direction of parallel projection is not parallel to view plane then we have oblique projection. There are two types of oblique projection cavalier projection and cabinet production.

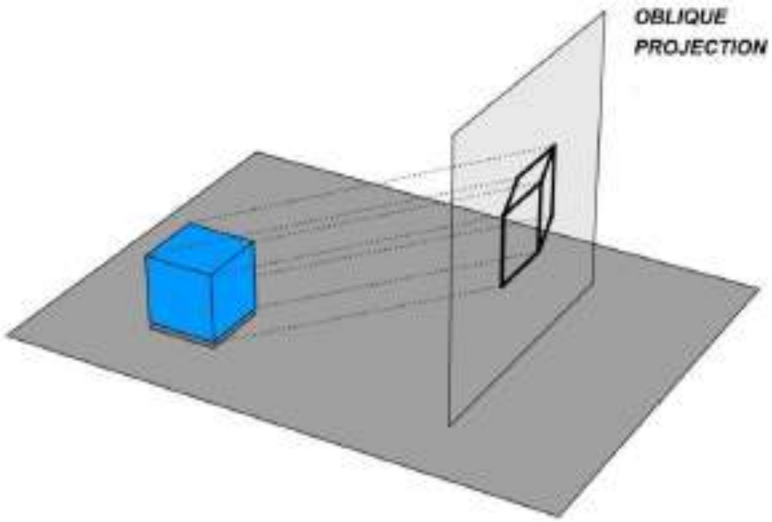


Fig 57 Oblique projection

1. Cavalier projection: Whenever the objects' edge parallel to the co-ordinate axis also will be parallel to the front face that is called as Cavalier projection in this direction of projection is slanted so the points with Z coordinates are positively project down and to the left on the view plane.
Z co-ordinate negatively projected up and right face. The angle of z axis can be any of the direction but the distance of the points shifted in a z direction must be equal to three dimensional view plane z distances that is called as Cavalier projection.

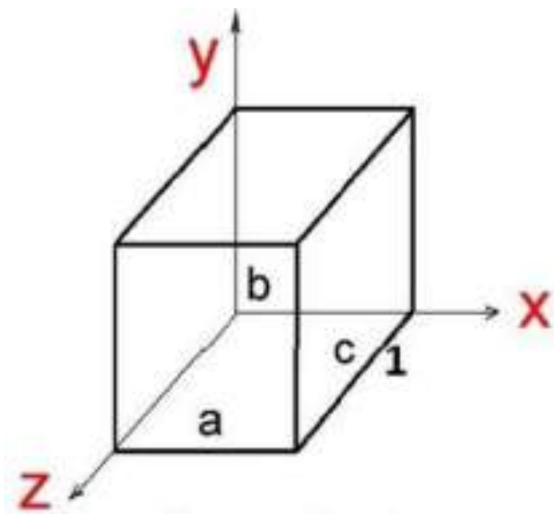


Fig 58 Cavalier Projection

2. Cabinet projection: In cabinet projection, the lines which are perpendicular to the projection plane are projected at one and half of the actual length of the object. The projection is said to be cabinet when an angle between projection plane and oblique projection is equals to 63.4□.

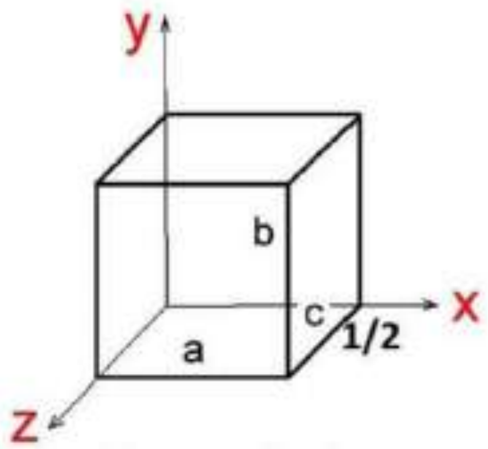


Fig 59 Cabinet Projection

ORTHOGRAPHIC PROJECTION

An orthographic projection is a type of parallel projection in which the projectors are parallel to each other and perpendicular to the plane of projection. It represents 3 dimensional objects in 2 dimensions. A square or rectangular viewing volume is formed for this projection by near, far, right, left, top and bottom clipping planes. Single orthographic projection does not provide sufficient information to reconstruct the shape of an image, therefore, multiple orthographic projection displays a single face of 3D object i.e. viewer can only see the one side of an object at a time.

In multiple projections front surface is parallel to plane of projection. Projection line of sights is perpendicular to the projection plane. Projectors are parallel to each other and originate from any point on object.

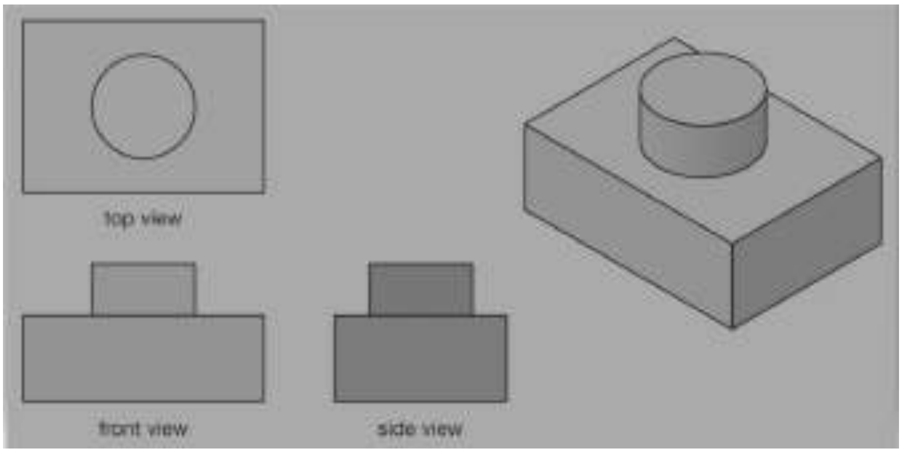


Fig 60 Orthographic Projection

VANISHING POINTS

A vanishing point is abstract point on the image plane where 2Dprojections of a set of parallel lines in 3D space appear to converge or intersect i.e. certain set of parallel lines appear to meet at different point which is known as vanishing point. When the set of parallel lines are perpendicular to a picture plane then the construction is known as onepoint perspective and their vanishing point corresponds to the “eye point” from which the image should be viewed for correct respective geometry.

Types of Vanishing point

1. One point perspective:

As the figure 6.14.1 shows end of the parallel line of hotels or houses after certain time of period the parallel lines meets each other at somepoint that point is called as one perspective vanishing point. Vanishing points are formed by the intersection parallel lines to one of the three axis x, y and z. all three parallel points are perpendicular to the same viewplane at single point.

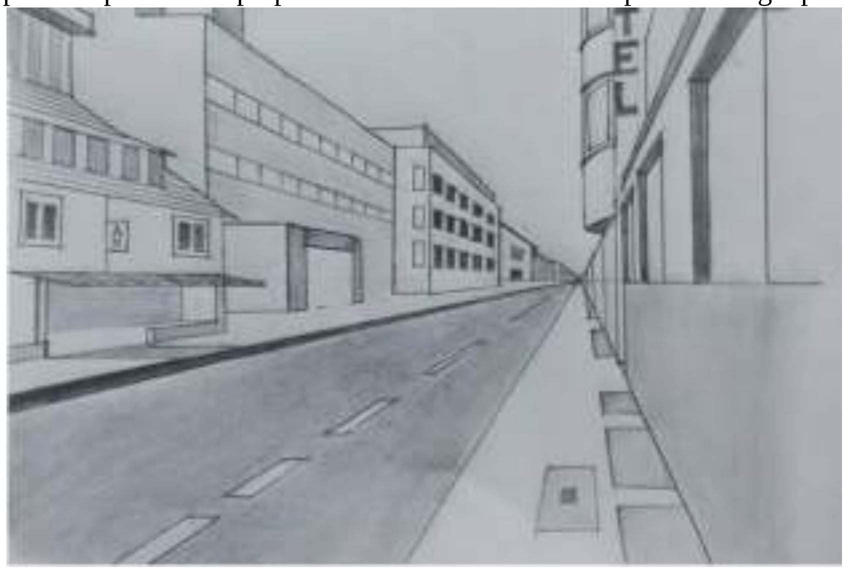


Fig 61 one point perspective vanishing point

2. Two point perspective:

In two points perspective side of an object vanish or recede towards one ofthe two vanishing points on the horizon.

Fig 62 Two point perspective vanishing point

3. Three point perspective:

In three point perspective, there are three vanishing points where two vanishing points are along the horizon and third one is located either above the horizon or below the horizon.

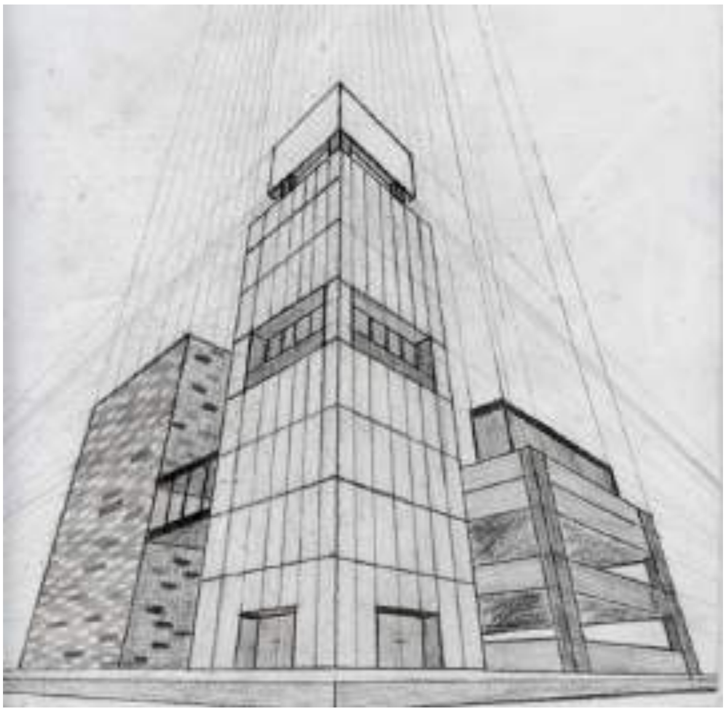


Fig 63 Three point perspective vanishing point

VIEW VOLUMES FOR PROJECTIONS

Converting 3D object into 2D is called projection. Projection operation convert the viewing plane coordinate into 2D coordinate system so that the object will look like same as it seen in the real world. There are two types of projection.

1. **Parallel projection**
2. **Perspective projection**

Parallel projection:

In this type of projection parallel line of object transforms in parallel lines only. a parallel projection is also called as relative projection as it preserves relative plane along with the parallel lines. Accurate view of object from multiple sides is view in this projection. This projection will not preserve realistic representation.

Parallel projection classification: 1 Orthographic Parallel projection 2.Oblique projection

Orthographic Parallel Projection: it is done by projecting points alongparallel lines that are perpendicular to the projection plane.

Oblique projection: oblique projections are obtained by projecting alongwith parallel lines that are NOT perpendicular to the projection plane

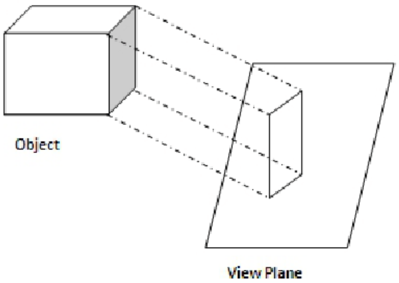


Fig 64 Parallel projection

Perspective projection:

Perspective projection transforms object positions to the view plane while converging to center point of projection. Perspective projection produces realistic views but does not preserve relative proportions. Projections of distant objects are smaller than the projections of object of the same size that are closer to the projection plane.

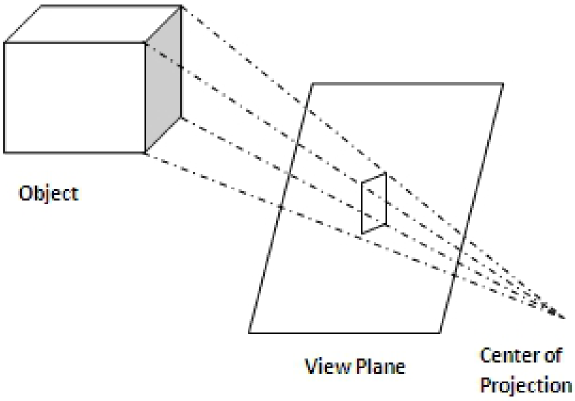


Fig 65 Perspective projection

View Volume:

A view volume is rectangular area in the view plane which controlshow much of the scene is viewed. T is formed by the window and the type of projection to be used. View volume is generally defined as a space between front and backplane.

Only those objects are seen which are within the window volume. The objects which are beyond the view volume it will be not seen. In view volume the perspective effects depends on the positioning of the center point of projection. If it is close to the view plane then the object look like closer and larger but when it is longer to the view plane it is look like smaller and far away from the view plane hence the size of the object affected by the relative position of the object. View volume is bounded by 6 planes left, right, top, bottom, front and back side.

STAGES IN 3D VIEWING

Viewing in 3D is divided mainly into following stages:

- Modelling transformation

- Viewing transformation
- Projection transformation
- Normalization transformation and Clipping
- View Port transformation

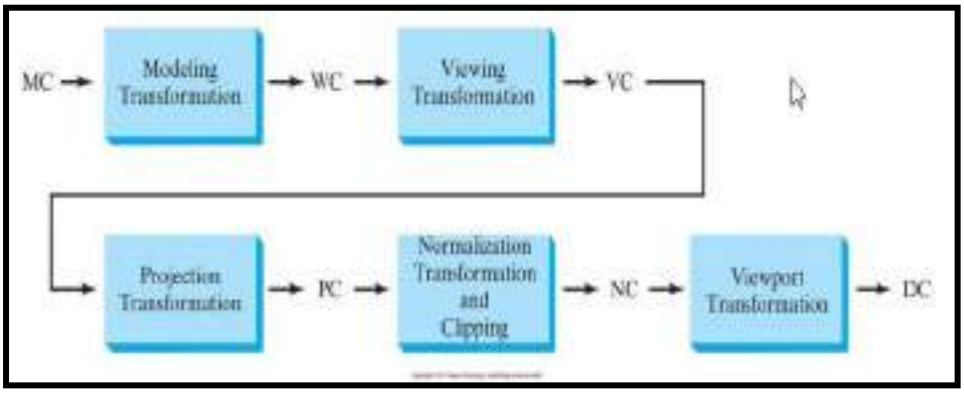


Fig: 66 Stages in 3D Viewing

1. **Modelling Transformation:**

   - Modelling Transformation can be obtained with the help of 3D transformations.
   - It transforms the object (Modelling) coordinates (MC) into 3D world coordinate system, will generate output as 3D world coordinates (WC). (3D to 3D).

2. **Viewing Transformation:**

   - The viewing-coordinate system is used in graphics packages as a reference for specifying the observer viewing position and the position of the projection plane.
   - It transforms the world coordinate into viewing coordinates (VC) by performing clipping against canonical view volume.

3. **Projection Transformation:**

   - Projection operation converts the viewing-coordinate description (3D) to coordinate positions on the projection plane (2D).
   - It projects the coordinates onto projection plane as Projection Coordinates (PC). (3D to 2D).

4. **Normalization Transformation and Clipping:**

   - The coordinates outside the view volume will get clipped and the transformation will get normalized and the Normalized Coordinates (NC) will be generated.

5. **Viewport Transformation:**

   - It maps the coordinate positions on the projection plane of the outputdevice.

- It transforms Normalized Coordinates into viewport is 2D Devicecoordinates (DC) or display.

CANONICAL VIEW VOLUME (CVV)

- Canonical, in computer graphics, is the standard state or behavior of an attribute. This term is used to refer to perceptions that are distinctive and/or natural.
- The view volume is instead mapped to a canonical view volume (CVV) which is a cube that extends from □1 to +1 in each dimension, having center at the origin. The dimensions of the CVV facilitate a fast and efficient clipping.

Example of CVV:

CVV is related to Projection transformation i.e. Orthographic (parallel) projection and Perspective projection as **shown in the below figure:**

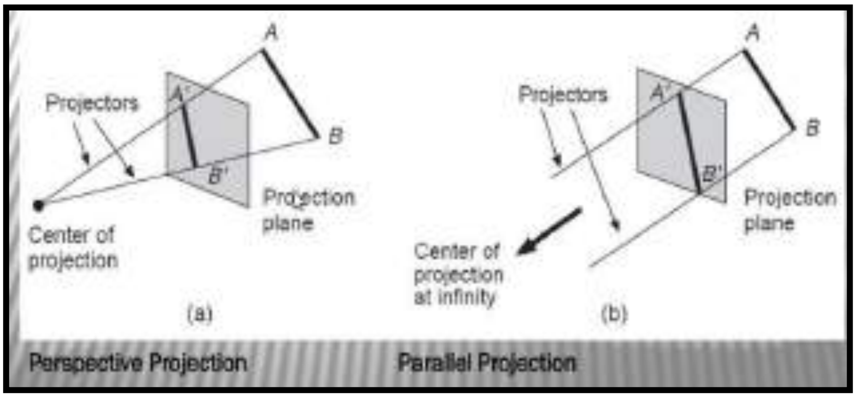


Fig 67 Perspective and Parallel (Orthographic) projection for CVV demonstrations

Note that **n** and **f** are typically given as distance which is always positive and because we are looking towards the **–z** direction, the actual coordinates become **–n and –f.**

Also note the changes in the z-direction. This makes objects further away from the camera to have larger z values. In other words, CVV is a left- handed coordinate system.

We need to map the box with corners at (l, b, -n) and (r, t, -f) to the (-1, -1, -1) and (1, 1, 1) of CVV.

SPECIFYING AN ARBITRARY 3D VIEW

- We can view the object from the side, or the top, or even from behind.
- Therefore, it is necessary to choose a particular view for a picture by first defining a view plane.
- A view plane is nothing but the film plane in a camera which is positioned and oriented for a particular shot of the scene.

- World coordinate positions in the scene are transformed to viewing coordinates, and then viewing coordinates are projected onto the view plane.
- The procedure to project from 3D to 2D given a finite view volume, will be as follows:
    - ❖ Apply a normalizing transform to get to the canonical view volume.
    - ❖ Clip against the canonical view volume
    - ❖ Project onto the view plane
    - ❖ Transform into viewport
- To specify an arbitrary view, we should be able to place the view plane anywhere in 3D. Specify the direction of the plane and its position in the world reference coordinate system (WRC).
    - ❖ A View Reference Point (VRP) which is to point on the plane.
    - ❖ A View Plane Normal (VPN) which is the normal vector to theplane.
    - ❖ A View Up Vector (VUP) which is a vector from which wedetermine which way is up.
- **Window on View plane:**
    - ❖ **PRP:** Projection Reference Point
    - ❖ **CW:** Center of Window
    - ❖ **COP:** Center of Projection (Perspective Projection)
    - ❖ **DOP:** Direction of Projection (Parallel Projection)
    - ❖ PRP and CW are used to determine COP and DOP
    - ❖ **Perspective:** COP = PRP
    - ❖ **Parallel:** DOP = PRP – CW
- **Coordinate Systems:**
    - ❖ **WC:** World Coordinates - normal, 3-space (x, y, z)
    - ❖ **VRC:** Viewing Reference Coordinates - defined by VRP, VPNand VUP

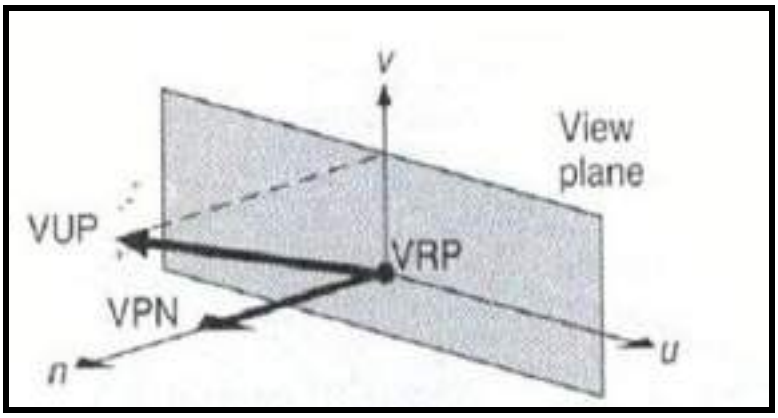


Fig 68: The view plane is defined by VPN and VRP, the v axis is defined by the projection of VUP along VPN onto the view plane. The u axis forms the right-handed VRC system with VPN and v.

- The notation (WC) or (VRC) is added to the table as a prompt of the coordinate system in which the viewing parameters is given. The form of the table is demonstrated for the default viewing specification used by PHIGS.

- The corresponding values for default view volume is given by,

Viewing parameter	**Value**	**Comments**
VRP(WC)	(0,0,0)	Origin
VPN(WC)	(0,0,1)	Z axis
VUP(WC)	(0,1,0)	Y axis
PRP(VRC)	(0.5,0.5,1.0)	
Window(VRC)	(0,1,0,1)	
Projection type	parallel	

Perspective Projections

- To obtain the front one-point perspective view of the house shown in **Figure** the position the center of projection (which can be thought of as the position of the viewer) at x = 8, y = 6, and = 84. The **x** value is selected to be at the horizontal center of the house and the **y** value to correspond to the estimated eye level of a viewer standing on the (x, z) plane; the zvalue is arbitrary. In this case, z is removed 30 units from the front of the house (z = 54 plane). The window has been made quite large, to guarantee that the house fits within the view volume. All other viewing parameters have their default values, so the overall set of viewing parameters is as follows:

VRP(WC)	(0, 0, 0)
VPN(WC)	(0, 0, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(8, 6, 84)
window(VRC)	(−50, 50, −50, 50)
projection type	perspective

- Although the image in **Figure** is indeed a perspective projection of the house, it is very small and is not centered on the view surface. We would prefer a more centered projection of the house that more nearly spans the entire view surface, as in **Figure** we can produce this effect more easily if the view plane and the front plane of the house coincide.

- Now, because the front of the house extends from 0 to 16 in both x and y, a window extending from 1 to 17 in x and y produces reasonable results.

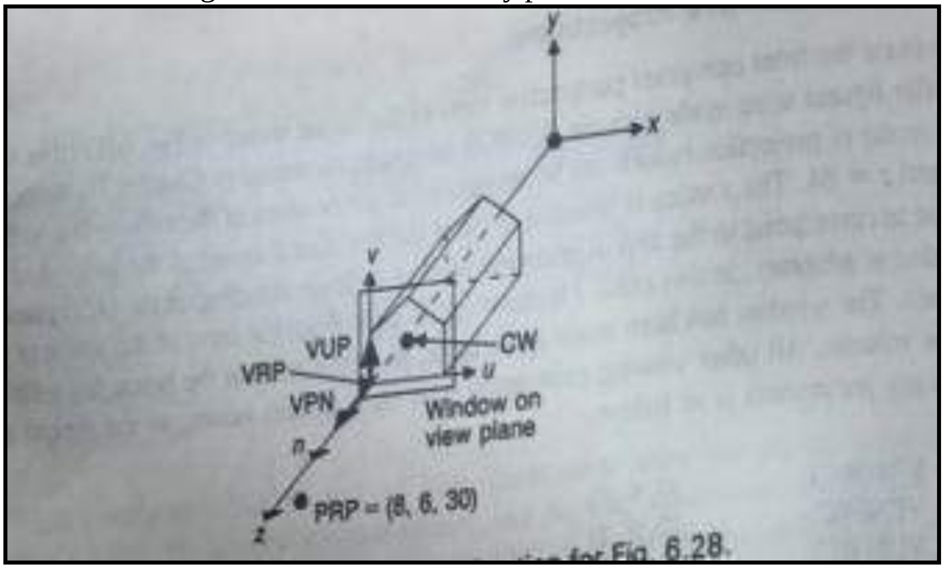


Fig 69 The viewing situation

Parallel Projections:

- We create a front parallel projection of the house (**Figure**) by making the direction of projection parallel to the z axis and direction of projection is determined by the PRP and by the center of the window.
- With the default VRC system and a window of (-1, 17,-1, 17), the center of the window is (8, 8, 0).
- The window is symmetrical about VRP, which implies that VRP, the originof the VRC system, is the window's center. Placing PRP as specified meansthat the direction of projection, which is the vector from PRP to the window center, is (0, 0, 0) - PRP= - PRP, or its negative PRP.

Finite View Volumes

- In all the example, the view volume has been expected to be infinite.
- The front and back clipping planes, help to determine a finite view volume.
- These planes, both of which are parallel to the view plane, are at distances F and B respectively from the view reference point, measured from VRP along VPN.
- To avoid a negative view volume, we must ensure that F is algebraically greater than B.

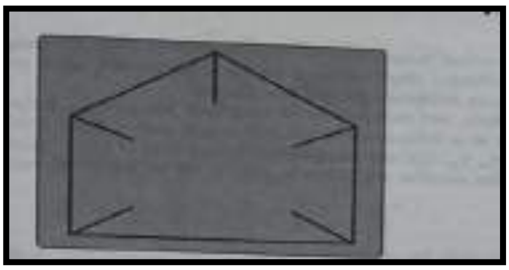


Fig 70 Perspective projection of the house with back clipping planeat z = 31.

- If a distance is given, then clipping against the corresponding plane is assumed; otherwise, it is not.
- If the front and back clipping planes are moved dynamically, the 3D structure object being viewed can often be discerned more readily than it can with a static view.

THE MATHEMATICS OF PLANAR GEOMETRICPROJECTIONS

- Planar projections are the subset of 3D graphical projections constructed by linearly mapping points in three-dimensional projectionplane. The projected point on the plane is chosen such that it is collinear with the corresponding three-dimensional point and the centre of projection. The lines connecting these points are commonly referred to as projectors.
- Type of planar geometric projection is obtained by the proper choice of a projection plane and a center of projection. The mathematical properties of each method of projection are related to the properties of related objects.
- Each type of projection provides variety of visual effects.
- Mathematically, planar projections are linear transformations actingon a point in three-dimensional space a(x, y, z) to give a point b (u, v) on the projection plane.
- These transformations consist of various compositions of the five transformations: **orthographic, projection, rotation, shear, translation and perspective.**
- For simplicity, we start by assuming that, in the perspective projection, the projection plane is normal to the z-axis at z=d, and that, in the parallel projection, the projection plane is the z=0 plane. Each of the projection can be defined by a 4 x 4 matrix. Because the projection matrix can be composed with transformation matrices, allowing two operations (transform, then project) to be represented as a single matrix.
- In arbitrary projection planes we derive 4 x 4 matrices for several projections, beginning with a projection plane at a distance d from the origin and a point **P** to be projected onto it.
    - This formulation allows d, the distance to the center of projectionto tend to infinity.
    - M_{per} applies only in the special case in which the center of projection is at the origin; M_{ort} applies only when the direction of projection is parallel to the z axis.

COMBINED TRANSFORMATION MATRICES FOR PROJECTIONS AND VIEWING

After having all required matrices of projection and viewing, we can now combine all our transformation into one overall matrix, which will converta point from world coordinates to Normalized Device Coordinates (NDC) while retaining a representation of the relative depth of the points.

MTOT=Awv * Mp * N

Where,

MTOT: - Represents Combined transformation matrices for projectionand viewing

Awv: - Represents Metrics of real-world coordinates **Mp: -**

Represents Metrics of perspective transformation**N: -**

Represents Normalization Matrix

This will give us the view volume in its final configuration, called as **Canonical View Volume.**

COORDINATE SYSTEMS AND MATRICES

Co-ordinate System:

1. Screen coordinates
2. Local co-ordinate system or 3D modelling coordinate
3. Word coordinate
4. View reference coordinates
5. Normalized projection coordinates
6. 2D device coordinate

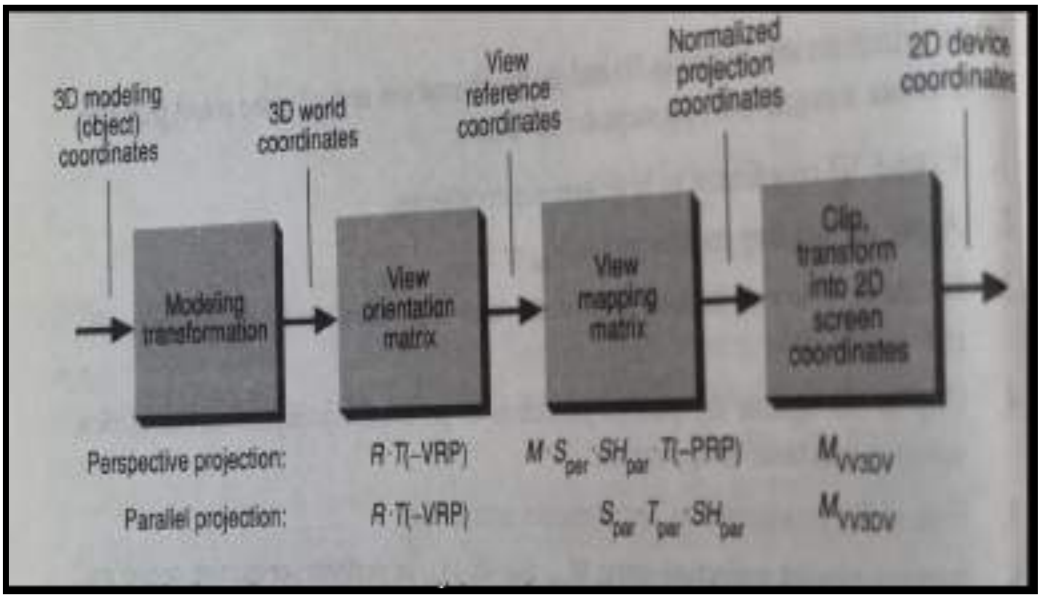


Fig 71 Coordinate systems and how they relate to one another.

The matrices underneath each stage effect the transformation appliedat that stage for the perspective and parallel projections.

- Starting with the coordinate system that is utmost detached from the real display device, on the left of **Figure**, individual objects are well-defined in an object-coordinate system, PHIGS **(Programmer's Hierarchical Interactive Graphics System)** calls this as modeling-coordinate system; the term local coordinate system is also commonly used.
- Objects are transformed into the world-coordinate system, the system in which a scene or complete object is represented in the computer, by the modeling transformation. This system is sometimes called the problem-coordinate system or application-coordinate system.
- The view-reference coordinate system is used by PHIGS as a coordinate system to define a view volume. It is also called the (u, v, n) system, or the (u, v, VPN) system.
- The Core system used alike, but unidentified, left-handed system. The left-handed system is used so that, with the eye or camera at the origin looking toward +z, increasing values of z are beyond away from the eye, x is to the right, and y is up.
- Other packages, such as Pixar's RenderMan [PIXA88], place constraints on the view-reference coordinate system, requiring that the origin be at the center of projection and that the view plane normal be the z axis known as eye-coordinate system, Renderman and some other systems use the term camera-coordinate system.
- The first three steps of the perspective-projection normalizing transformation convert from the world-coordinate system into the eye-coordinate system. The eye coordinate system is sometimes left- handed.
- From eye coordinates, we next go to the normalized-projection coordinate system, or 3D screen coordinates, the coordinate system of the parallel-projection canonical view volume (and of the perspective- projection canonical view volume after the perspective transformation).
- The Basic system calls this system 3D normalized device coordinates. Sometimes, the system is called 3D logical device coordinates. The term normalized generally means that all the coordinate values are in either the interval [0, 1] or 1-1, 1], whereas the term logical generally means that coordinate values are in some other prespecified range,such as [0, 1023], which is naturally defined to resemble some extensively available device's coordinate system.

- Projecting from 3D into 2D known as 2D device coordinate system, also called the normalized device-coordinate system, the image- coordinate system or the screen-coordinate system stated by RenderMan.
- Other terms used include screen coordinates, device coordinates, 2D device coordinates, and physical device coordinates (in contrast to the logical device coordinates stated earlier).
- RenderMan calls the physical fore of the space as raster coordinates.

CAMERA MODEL AND VIEWING PYRAMID

- In 3D computer graphics, the view frustum (also called viewingfrustum) is the region of space in the modeled world that may appear on the screen; it is the field of view of the "notional camera".
- The view frustum is typically obtained by taking a frustum- that is a truncation with parallel planes- of the pyramid of vision, which is theadaption of (idealized) cone of vision that a camera or eye would have to the rectangular viewports typically used in computer graphics.
- The exact shape of this region varies depending on what kind camera lens is being simulated, but typically it is a frustum of a rectangular pyramid (hence the name). The planes that cut the frustum perpendicular to the viewing direction are called the near plane and the far plane. Objects closer to the camera than the near plane or beyond the far plane are not drawn. Sometimes, the far plane is placed infinitely far away from the camera so all objects within the frustum are drawn regardless of their distance from the camera.

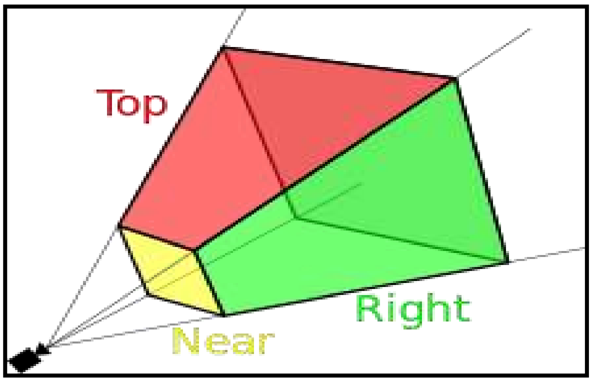


Fig 72 View frustum

View frustum culling is the process of removing objects that lie completely outside the viewing frustum from the rendering process. Rendering these objects would be a waste of time since they are not directly visible. To make culling fast, it is usually done using boundingvolumes surrounding the objects rather than the objects themselves.

Module IV: Hidden surface elimination algorithms
4.1 Z-buffer.
4.2 Scan-line.
4.3 Sub-division.
4.4 Painter's algorithm.
4.5 Illumination Models: Diffuse reflection, Specular reflection, refracted light, texture surface patterns, Halftoning, Dithering.
4.6 Surface Rendering Methods: Constant Intensity method, Gouraud Shading, Phong Shading.

VISIBLE-SURFACE DETERMINATION

Techniques for efficient Visible-surface Algorithms

The techniques to perform visible surface algorithms efficiently are discussed in the following sections.

1. Coherence

The coherence is defined as the degree to which parts of an environment, or its projection exhibit local similarities. Such as similarities in depth, colour, texture and so on. To make algorithms more efficient we can exploit these similarities when we reuse calculations made for one part of the environment or a picture for other nearby parts, either without changes or with some incremental changes. Different kinds of coherence we canuse in visible surface algorithms are.

i. **Object coherence**: if one object entirely separates from another, comparisons may need to be done only between the two objects, and not between their components faces or edges.

ii. **Face coherence:** Usually surface properties vary smoothly across a face. This allows the computations for one part of face to be used withincremental changes to the other parts of the face.

iii. **Edge coherence:** The visibility of edge may change only when it crosses a visible edge or penetrates a visible face.

iv. **Implied edge coherence:** If one planar face penetrates another their line of intersection can be determined from two points of intersection.

v. **Area coherence:** A group of adjacent pixels is often belonging to the same visible face.

vi. Span coherence: It refers to a visibility of face over a span of adjacent pixels on a scan line. It is special case area coherence.

vii. **Scan line coherence:** The set of visible object spans determined for one scan line of an image typically changes very little from the set on the previous line.

viii. **Depth coherence:** Adjacent parts of the same surface are typically same or very close depth. Therefore, once the depth at one point of the surface is determined, the depth of the points on the rest of the surfacecan be determined by at the simplest incremental calculation.

ix. **Frame coherence:** Pictures of the same scene at two successive points in time are likely to be quite similar, except small changes in object and view ports. Therefore, the calculations made for one picture can be reused for the next picture in as sequence.

2. Perspective Transformation:

Visible-surface determination is done in a 3D space prior to the projection into 2d that destroys the depth information needed for depth comparisons, and depth comparisons are typically done after the normalizing transformation.

Due to this projector are parallel to the Z axis in parallel projections or emanate from the origin in perspective projections.

In parallel projection, when x1=x2 and y1=y2 we can say that points are on the same projector.

However, in perspective projection we must perform four divisions: x1/z1=x2/z2 and y1/z1=y2/z2 to determine whether the points are on the same projector.

These divisions can be avoided by first transforming a 3D object into the 3D screen-coordinate systems, so that the parallel projection of the transformed object is the same as the perspective projection of the untransformed object.

3. Extents and bounding Volumes:

a. Two objects with their projections and the rectangular screen extents surrounding the projections.

b. It is easier to check the overlapping of extents than the projections,and we can say that when the extents are not overlapping then projections are also not overlapping. Therefore, extents must be compared first and then the projections must be compared only if the extents are overlapped. This avoids unnecessary comparisons in checking the overlapping of projections.

c. Extents used to surround the object themselves rather than their projections; in this case, the extent become solid and are commonly known as bounding volumes. In this case extent can be used to check whether two objects are overlapped or not.

d. Extent and bounding volumes are used to determine whether projectorintersects an object or not.

4. Back-Face Culling

a. When an object is approximated by a solid polyhedron, its polygonal faces completely enclose its volume. In such case, if none of the polyhedron’s interior is exposed by the front clipping plane, then those polygons whose surface normal point away from the observer lie on a part of the polyhedron whose visibility is completely blocked by other closer polygons.

b. Polygons (A, B, D, F) in gray are eliminated, whereas front-facing polygon (C, E,G,H) are retained.

c. These invisible back facing polygons can be eliminated from further processing. Such a technique to eliminate the back facing polygon from further processing is known as Back-face Culling.

5. Spatial Partitioning:

a. In this technique, subdivision rule is applied to break down a large problem into a number of smaller ones. This object and their projections are assigned to spatially coherent groups as a pre-processing step. This partitioning speed up the process of determining which object intersect with a projector. Because now it is necessary to test only the objects lying within the partitions which intersect with projector.

b. When objects are unequally distributed in space the adaptive partitioning is more suitable. Because it allows variable size of each partition.

6. Hierarchy

a. In hierarchical structure different levels are assigned to the object and there is a parent-child relationship between the objects.

b. In this structure, each child is considered as a part of its parent. This allows to restrict the number of object comparison needed by a visible-surface algorithm.

c. If the parent level object is fail to intersect, the lower level objects belongs to the parent do not need to be tested for intersection.

Object Space Method and Image Space Method:

Object Space method	**Image Space method**
1. It deals with object definition directly	1. It is a pixel-based method. It is concerned with the final image, what is visible within each raster pixel.
2. Here surface visibility is determined.	2. Here line visibility or point visibility is determined.
3. It is performed at the precision with which each object is defined, No resolution is considered.	3. It is performed using the resolution of the display device.
4. Calculations are not based on the resolution of the display so change of object can be easily adjusted.	4. Calculations are resolution base, so the change is difficult to adjust.
5. These were developed for vector graphics system.	5. These are developed for raster devices.
6. Object-based algorithms operate on continuous object data.	6. These operate on object data.
7. Vector display used for object method has large address space.	7. Raster systems used for image space methods have limited address space.
8. Object precision is used for application where speed is required.	8. There are suitable for application where accuracy is required.
9. It requires a lot of calculations if the image is to enlarge.	9. Image can be enlarged without losing accuracy.

10. If the number of objects in the scene increases computation time also increases.	10. In this method complexity increase with the complexity of visible parts.

CATEGORIES OF ALGORITHMS

Back- face removal:

1. A polygon has two surfaces, a front and a back just as a piece of paper does. We might picture our polygons with one side painted light and the other painted dark. But the question is "how to find which surface is light or dark".
2. When we are looking at the light surface, the polygon will appear to be drawn with counter clockwise pen motions, and when we are looking at the dark surface the polygon will appear to be drawn with clockwise pen motions.
3. Let us assume that all solid objects are to be constructed out of polygons in such a way that only the light surfaces are open to the air; the dark faces meet the material inside the object. This means that when we look at an object face from the outside, it will appear to be drawn counter clockwise.
4. If a polygon is visible, the light surface should face towards us and the dark surface should face away from us. Therefore, if the direction of the light face is pointing towards the viewer, the face is visible, otherwise the face is hidden and should be removed.
5. The direction of the light face can be identified by examining the result

   $N.V > 0$

   Where N: Normal vector to the polygon surface with cartesian components(A,B,C).

   1V: A vector in the viewing direction from the eye position.
6. We know that, the dot product of two vector gives the product of the length of the two vectors times the cosine of the angle betweenthem.
7. This cosine factor is important to us because if the vectors are in the same direction ($0 \leq \Theta < \pi/2$), then the cosine is positive, and the overall dot product is positive. However, if the direction is opposite ($\pi/2 < \Theta \leq \pi$), then the cosine and the overall dot product is negative.
8. If the dot product is positive, we can say that the polygon faces towards the viewer; otherwise it faces away and should be removed.
9. In case, if object description has been converted to projection coordinates and our viewing direction is parallel to the viewing zv axis, then $V=(0, 0, V_z)$ and $V.N = V_z C$
10. So that we only have to consider the sign of C, the Z component of the normal vector N. Now, if the z component is positive, then the polygon faces towards the viewer, if negative, it faces away.

Scan Line Algorithm:

A scan line method of hidden surface removal is another approach of image space method. It is an extension of the scan line algorithm for filling polygon interiors. Here, the algorithm deals with more than one surfaces. As each scan line is processed, it examines all polygon surfaces intersecting that line to determine which are visible. It does the depth calculation and finds which polygon is nearest to the view plane. Finally,it enters the intensity value of the nearest value of the nearest polygon at that position into the frame buffer.

1. We know that scan line algorithm maintains the active edge list. This active edge list contains only edges that cross the current scan line, sorted in order of increasing x. The scan to indicate whether a positionalong a scan line is inside or outside of the surface. Scan lines are processed from left to right. At the leftmost boundary of a surface, the surface flag is turned ON, and at the rightmost boundary, it is turned OFF.

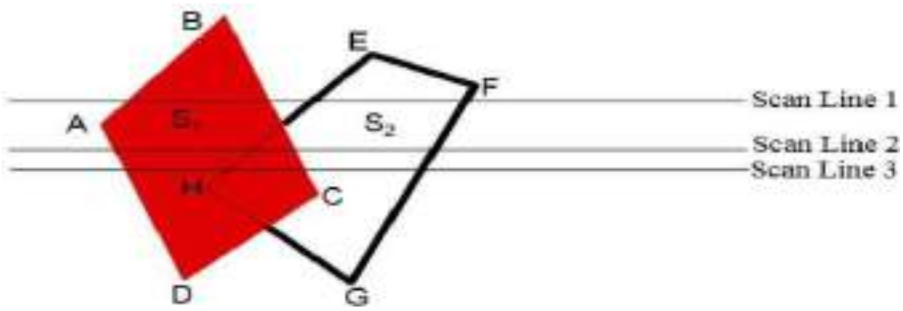


Fig 73: Example 1 Scan Line Algorithm

2. **Example 1**: The figure illustrates the scan line method for hidden surface removal. The active edge list for scan line 1 contains the information for edges AD, BC, EH an FG. For the positions along this scan line between edges AD and BC, only the flag for surface S_1 is ON. Therefore, no depth calculations are necessary, and intensity information for surface S_1 is entered the frame buffer. Similarly, between edges EH and FG, only the flag for surface S_2 is ON and during that portion of scan line intensity information for surface S_2 is entered the frame buffer.

3. For scan line, the active edge list contains edges AD, EH, BC, FG. Along the scan line 2 from edge AD to edge EH, only the flag for surface S_1 is ON. However, between edges EH and BC, the flags for both surface are ON. In this portion of scan, line 2, the depth calculations are necessary. Here we have assumed that the depth of S_1 is less than the depth of S_2 and hence the intensities of surface S_1 are loaded into the frame buffer. Then for edge BC to edge FG portion of scan line 2 intensities of surface S_2 are entered the frame buffer because buffer during that portion only flag for S_2 is ON.

4. This algorithm is based on the Image-space method and concept of coherence. As its name suggests itself Scan-line algorithm, so it processes one line at a time rather than processing one pixel(a point on raster display) at a time.

5. **Edge list table**: This list maintains the record of all the edges by storing their endpoint coordinates. The x-coordinate that we choose, whose Y-coordinate = Ymin.

6. **Active edges table(list):** This table contains all those edges of the polygon that are intersected(crossed) by the current scan-line. The edges are dropped into the table in a sorted manner(Increasing value of x).

7. **Polygon table(list)**: This list consists of:
   a. Polygon Id.
   b. Plane equation.
   c. Color Information of the surface.
   d. Flag of surface(on/off)].

1. **First, examine the scanline(S_1), whose,** Active edge table (Aet) contains [AD,BC,RS,PQ], and the flag is set to "on" for surface(S_1). In addition it contains surface(S_2) and the flag set "off" for the region between BX and RX as it's an outer region of the polygon's surface and not to be projected at view-port(display devices). Now drop the color-intensities of the corresponding surfaces into the frame buffer(refresh buffer).

2. **Then, process the scanline(S_2), whose,** Active edge table (Aet) contains[AD,BC,RS,PQ], and the flag is set to "on" for surface (ABCD) and surface(PQRS). Both of the polygons surfaces are overlapping each other. So for this overlapped region that which of the surface intensity should be taken into account? calculate the depth (Z_{min}) of both surface(S_1) and surface(S_2) of the overlapped portion, ofsurface intensity . Next, If depth(S_1)>depth(S_2), then the Flag of surface S_1=" on" and intensityof surface S_1 will be considered else S_2, now Drop the color-intensities of the corresponding surfaces whose flag is set to on into the frame buffer(refresh buffer).

3. As Scanline(S_3) is passing through the same portion from where Scanline(S_2) is passing, S_3 also has the same Active edge table(Aet) components as S_2 has and no need to calculate the depth(S1) and depth(S2) again so S_3 can take the advantage of the concept of Coherence.

Z-Buffer Algorithm:

1. One of the simplest and commonly use image space approach to eliminate hidden surface is the **Z-buffer** or **depth buffer** algorithm. It is developed by Catmull. This algorithm compares surface depths at each pixel position on the projection plane.

2. The surface depth is measured from the view plane along the z axis of a viewing system. When object description is converted to projection coordinates (x, y, z), each pixel position on the view plane is specified by x and y coordinate, and z values gives the depth information. Thus,object depths can be compared by comparing the z-values.

3. The Z-buffer algorithm is usually implemented in the normalized coordinates, so that z values range from 0 at the back-clipping plane to 1 at the front clipping plane. The implementation requires another buffer called Z-buffer along with the frame buffer memory required for raster display devices. A Z-buffer is used to store depth values for each (x, y) position as surfaces are processed, and the frame buffer stores the intensity values for each position.
4. At the beginning the z-value at the back-clipping plane, and the frame buffer is initialized to the background colour. Each surface listed in the display file is then processed, one scan line at a time calculatingthe depth (z-value) at each (x, y) pixel position.
5. The calculated depth value is compared to the value previously stored in the Z-buffer at that position. If the calculated depth values is greaterthat the value stored in the Z-buffer, the new depth value is stored, and the surface intensity at that position is determined and placed in the same xy location in the frame buffer.
6. For example, among three surfaces, surface S_1 has the smallest depth at view position (x, y) and the highest z value. So, it is visible at that position.

Algorithm:

1. Initialize the Z-buffer and frame buffer so that for all buffer

   positionsz-buffer(x,y)=0 and Frame-buffer(x,y)=Ibackground
2. During scan conversion process, for each position on each polygon surface, compare depth values to previously stored values in the depth buffer to determine visibility.
3. Calculate z-value for each (x,y) position on the

   polygonIf z>z-buffer(x,y) then

   z-buffer(x,y)=z,

   Frame-buffer(x,y)= Isurface(x,y).
4. Stop.

   (After processing of all surfaces, the z-buffer contains depth valuesfor the

   visible surfaces and the frame buffer contain the correspondingintensity

   values for those surfaces.)

Advantages:

1. It is easy to implement
2. It can be implemented in hardware to overcome the speed problem.
3. Since the algorithm processes objects one at a time, the total numberof polygons in a picture can be arbitrarily large.

Disadvantages:

1. It requires an additional buffer and hence the large memory.
2. It is time consuming process as it requires comparison for each pixelinstead of for the entire polygon.

Warnock's Algorithm (Area Subdivision Algorithm):

An interesting approach to the hidden-surface problem was developed by Warnock. He Developed area subdivision algorithm which subdivides each area into four equal squares.

At each stage in the recursive-subdivision process, the relationship between projection of each polygon and the area of interest is checked for four possible relationships:

1. Surrounding polygon – One that completely encloses the (shaded)area of interest.
2. Overlapping or Intersecting Polygon –One that is partly inside andpartly outside the area.
3. Inside or Contained Polygon – One that is completely inside the area.
4. Outside or Disjoint Polygon- One that is completely outside the area.

After checking four relationship we can handle each relationship as follows:

1. If all the polygons are disjoint from the area, then the background colour is displayed in the area.
2. If there is only one intersecting or only one contained polygon, then the area is first filled with background colour, and then the part of the polygon contained in the area is filled with colour of polygon.
3. If there is a single surrounding polygon, but no intersecting or contained polygons, then the area is filled with the colour of the surrounding polygon.
4. If there are more than one polygon intersecting, contained in, or surrounding the area then we must do some more processing.

Algorithm:

1. Initialize the area to be the whole screen.
2. Create the list of polygons by sorting them with their z-values of vertices. Don't include disjoint polygons in the list because they are not visible.
3. Find the relationship of each polygon.
4. Perform the visibility decision test
   a. If all the polygons are disjoint from the area, then fill area with background colour.
   b. If there is only one intersecting or only one contained polygon

then first fill entire area with background colour and then fill the part of the polygon contained in the area with the colour of polygon.

c. If there is a single surrounding polygon, but no intersecting or contained polygons, then fill the area with the colour of the surrounding polygon.

d. If the surrounding polygon is closer to the viewpoint than all other polygons, so that all other polygons are hidden by it, fill the area with the colour of the surrounding polygon.

e. If the area is the pixel (x, y) and neither a, b, c, nor d apply, compute the z coordinate at pixel (x, y) of all polygons in the list. The pixel is then set to colour of the polygon which is closer tothe viewpoint.

5. If none of the above tests are true, then subdivide the area and go to step 2.

Advantages:

1. It follows the divide-and-conquer strategy; therefore, parallelcomputers can be used to speed up the process.
2. Extra memory buffer is not required.

Binary Space Partition (BSP) Trees:

Binary space partitioning is used to calculate visibility. To build the BSP trees, one should start with polygons and label all the edges. Dealing with only one edge at a time, extend each edge so that it splits the plane into two. Place the first edge in the tree as root. Add subsequent edges based on whether they are inside or outside. Edges that span the extension of an edge that is already in the tree are split into two and both are added to the tree.

Painter’s Algorithm (Depth Sorting Method):

- Depth sorting method uses both image space and object-space operations. The depth-sorting method performs two basic functions −
- First, the surfaces are sorted in order of decreasing depth.
- Second, the surfaces are scan-converted in order, starting with the surface of greatest depth.
- The scan conversion of the polygon surfaces is performed in image space. This method for solving the hidden-surface problem is often referred to as the painter's algorithm. The following figure shows the effect of depth sorting −The algorithm begins by sorting by depth. For example, the initial “depth” estimate of a polygon may be taken to be the closest z value of any vertex of the polygon.
- Let us take the polygon P at the end of the list. Consider all polygons Q whose z-extents overlap P’s. Before drawing P, we

make the following tests. If any of the following tests is positive, then we can assume P can be drawn before Q.

Visible-Surface Ray Tracing:

- Ray tracing is an image-based algorithm. For every pixel in the image,a ray is cast from the center of projection through that pixel and into the scene. The colour of the pixel is set to the colour of the object that is first encountered.

COMPARISON OF THE METHODS

- Hardware available? Use depth-buffer, possibly in combination withback-face elimination or depth-sort for part of scene.
- If not, choose dependent on complexity scene and type of objects:
- Simple scene, few objects: depth-sort
- Quadratic surfaces: ray-casting
- Otherwise: depth-buffer
- Many additional methods to boost performance (kD-trees, scene decomposition, etc.)