



S.Y.B.Sc. (IT)
SEMESTER - IV (CBCS)

**COMPUTER GRAPHICS AND
ANIMATION**

SUBJECT CODE : USIT405

Prof. Suhas Pednekar

Vice Chancellor

University of Mumbai, Mumbai.

Prof. Ravindra D. Kulkarni

Pro Vice-Chancellor,
University of Mumbai.

Prof. Prakash Mahanwar

Director
IDOL, University of Mumbai.

- Programe Co-ordinator** : **Shri. Mandar Bhanushe**
Head, Faculty of Science and Technology,
IDOL, Univeristy of Mumbai
- Course Co-ordinator** : **Ms. Gouri Sawant**
Assistant Professor B.Sc.I.T,
IDOL, University of Mumbai.
- Editor** : **Dr. Ghayathri**
Associate Professor in Computer Science (PG),
Kongu Arts and Science College (Autonomus)
- Course Writers:**
- : **Ms. Binita Thakkar**
Assistant Professor,
VIVA College of Arts, Commerce and Science, Virar W
 - : **Ms. Rohini Desai**
Assistant Professor, Vidyalankar School of Information Technology
 - : **Ms. Priti Pathak**
Assistant Professor, Mulund College of Commerce,
Sarojni Naidu Road, Mulund(West), Mumbai- 80
 - : **Ms. Anjali Gaikwad**
Assistant Professor, JES college of commerce, science and
Information technology
 - : **Ms. Kshitija Palav**
Assistant Professor, VIVA College of Arts Commerce and Science.
Virar(W). Tal: Vasai, Dist: Palghar

December 2021, Print I

Published by

Director

Institute of Distance and Open Learning, University of Mumbai, Vidyanagari, Mumbai - 400 098.

DTP COMPOSED AND PRINTED BY

Mumbai University Press,
Vidyanagari, Santacruz (E), Mumbai - 400098.

CONTENTS

Chapter No.	Title	Page No
Unit I		
1.	Introduction To Computer Graphics	1
2.	Display Technologies	15
3.	Scan Conversion Line And Circle	29
4.	Scan Conversion – Line And Polygon Clipping	45
Unit II		
5.	Two Dimensional Transformations	57
6.	Three Dimensional Transformations	75
Unit III		
7.	Viewing In 3d	99
8.	Light: - Light: Radiometry, Transport, Equation, Photometry	127
9.	Color: - Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance	143
Unit IV		
10.	Visible-Surface Determination And Plane Curves And Surfaces	157
11.	Plane Curves And Surfaces	171
Unit V		
12.	Computer Animation	186
13.	Image Manipulation And Storage	199

Syllabus

F.Y.B. Sc. (Information Technology)		Semester – IV	
Course Name: Computer Graphics and Animation		Course Code: USIT405	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	-	25

Unit	Details	Lectures
I	<p>Introduction to Computer Graphics: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays.</p> <p>Scan conversion: Digital Differential Analyzer (DDA) algorithm, Bresenham's Line drawing algorithm. Bresenham's method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Clipping Lines algorithms– Cyrus-Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.</p>	12
II	<p>Two-Dimensional Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations.</p>	12

	<p>Three-Dimensional Transformations: Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.</p>	
III	<p>Viewing in 3D Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, The Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid. Light: Radiometry, Transport, Equation, Photometry Color: Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance</p>	12
IV	<p>Visible-Surface Determination: Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods. Plane Curves and Surfaces: Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Parametric Representation of an Ellipse, Parametric Representation of a Parabola, Parametric Representation of a Hyperbola, Representation of Space Curves, Cubic Splines, Bezier Curves, B-spline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces.</p>	12
V	<p>Computer Animation: Principles of Animation, Key framing,</p>	

	Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects. Image Manipulation and Storage: What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.	12
--	--	-----------

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Computer Graphics - Principles and Practice	J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes	Pearson	2nd	
2.	Steve Marschner, Peter Shirley	Fundamentals of Computer Graphics	CRC press	4th	2016
3.	Computer Graphics	Hearn, Baker	Pearson	2nd	
4.	Principles of Interactive Computer Graphics	William M. Newman and Robert F. Sproull	TMH	2nd	
5.	Mathematical Elements for CG	D. F. Rogers, J. A. Adams	TMH	2nd	2008

UNIT I

1

INTRODUCTION TO COMPUTER GRAPHICS

Unit Structure

- 1.0 Objectives
- 1.1 Overview of Computer Graphics
 - 1.1.1 Definitions
 - 1.1.2 Advantages of Computer Graphics
- 1.2 Computer Graphics Applications
 - 1.2.1 Computer Aided Design
 - 1.2.2 Computer Art
 - 1.2.3 Entertainment
 - 1.2.4 Education and Training
 - 1.2.5 Presentation Graphics
 - 1.2.6 Visualization
 - 1.2.7 Image Processing
 - 1.2.8 Graphical User Interfaces
- 1.3 Computer Graphics Software
 - 1.3.1 Coordinate Representations
 - 1.3.2 Graphic Functions
 - 1.3.3 Software Standards
- 1.4 Input Devices
 - 1.4.1 Keyboards
 - 1.4.2 Mouse
 - 1.4.3 Trackball and Spaceball
 - 1.4.4 Joysticks
 - 1.4.5 Data Glove
 - 1.4.6 Image Scanner
 - 1.4.7 Touch Panels
 - 1.4.8 Light Pens
 - 1.4.9 Voice Systems
- 1.5 Active and Passive Graphics
- 1.6 Summary
- 1.7 Unit End Exercise
- 1.8 References
- 1.9 Bibliography

1.0 OBJECTIVES

After this chapter, student will be able to:

- Define computer graphics
- State various applications of graphics
- Understand coordinate system and various graphics functions
- Identify various input devices used in graphics

1.1 OVERVIEW OF COMPUTER GRAPHICS

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

1.1.1 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 and ~~number of pixels~~ 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.

1.1.2 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.

1.2 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

1.2.1 Computer Aided Design :

It is used in

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

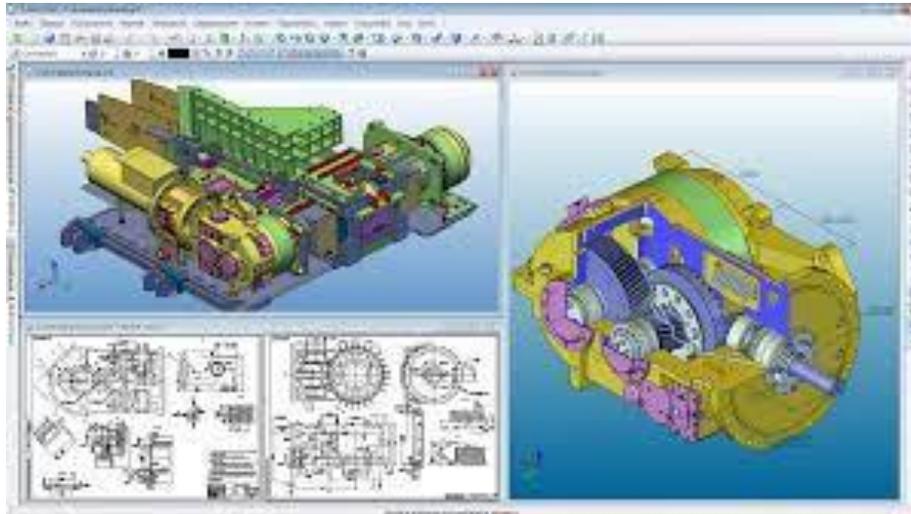


Fig. 1 Use of CAD in computer graphics

1.2.2 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

1.2.3 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, animated shows or TV shows.

- Games can also be made with interactivity features.



Fig. 3 Use of Entertainment in computer graphics

1.2.4 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.

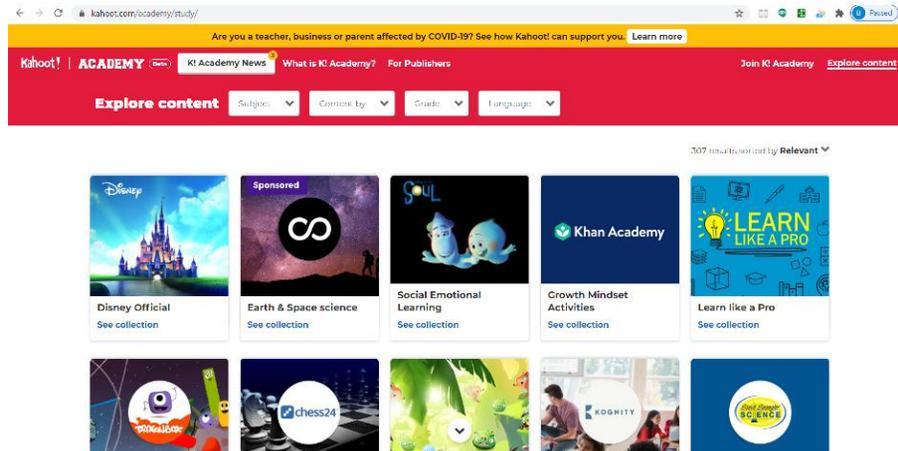


Fig. 4 Use of Kahoot in Education

1.2.5 Presentation Graphics :

- Reports can be generated on slides.
- It is used for summarizing mathematical, financial, statistical and scientific 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

1.2.6 Visualization :

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



Fig. 6 Use of Visualization in computer graphics

1.2.7 Image Processing :

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

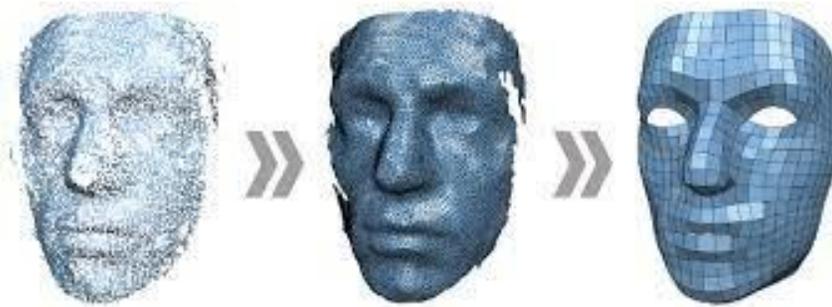


Fig. 7 Use of Image Processing in computer graphics

1.2.8 Graphical User Interfaces :

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

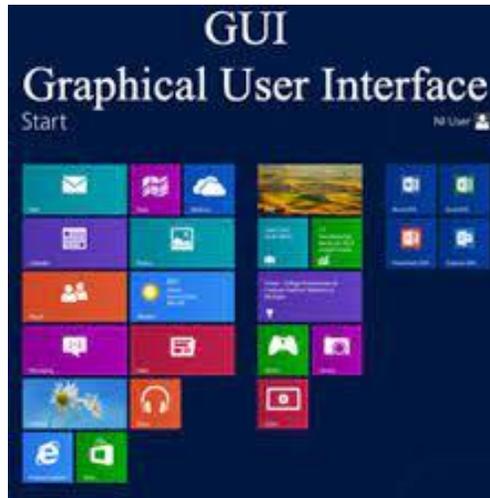


Fig. 8 Use of GUI in computer graphics

1.3 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.

1.3.1 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 display a 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 used as a workstation.

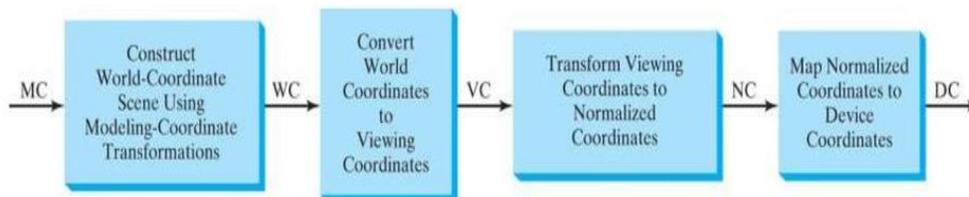


Fig. 9 Viewing Pipeline

1.3.2 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, orientation or 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.

1.3.3 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 first graphics 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.

1.4 INPUT DEVICES

1.4.1 Keyboards:

- To enter any kind of string or number, an alphanumeric keyboard is used 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

1.4.2 Mouse:

- To position the cursor on the screen, mouse is used.
- The roller present at the bottom of the mouse records the direction of the 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

1.4.3 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

1.4.4 Joysticks:

- A joystick consists of a small, vertical lever (called the stick) mounted on a base that is used to move the screen cursor in all directions.
- Position on the screen is selected with actual stick movement or by applying 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

1.4.5 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

1.4.6 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 screen area.

- For scanned text input, various editing operations can be performed on the 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

1.4.7 Touch Panels :

- A type of display screen that has a touch-sensitive transparent panel covering 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 of using mouse and keyboards.
- Generally used on smart phones and laptops.



Fig. 16 Touch Panel

1.4.8 Light Pens :

- It is a pencil-shaped device used to select screen positions by detecting the light coming from points on the CRT screen.
- Also used to select menu driven applications or draw pictures on monitor 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 to CPU.



Fig. 17 Light Pens

1.4.9 Voice Systems:

- It is used in some graphics workstations as input devices to accept voice 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

1.5 ACTIVE AND PASSIVE GRAPHICS

Active Graphics	Passive Graphics
Also known as interactive graphics.	Also known as non-interactive graphics.
In active graphics, users have control over the picture displayed on the screen.	In passive graphics, users do not have any control over the picture displayed on the screen.
These graphics include animations.	These graphics do not change its shape on display.
It is a two-way communication between the computer and the user.	It is a one-way communication between the computer and the user.

Low cost.	High cost.
Image quality produced is high.	Image quality produced is not too good.
E.g., video games, drawing on touch screens	E.g., titles on TV screen, data representation through charts

1.6 SUMMARY

- Computer graphics is a way in which pictures or object can be created and manipulated.
- It can be used in creation of 2D images, 3D images or animations.
- The applications of computer graphics can be in the field of education, training, CAD systems, entertainment, presentations, computer art, image processing and many more.
- The use of various graphics functions with graphics standard make up a graphics software.
- Various devices like keyboard, mouse, data glove, joysticks, etc. can be used to provide input to any graphics system.

1.9 UNIT END EXERCISE

1. Define computer graphics.
2. List and explain any five applications of computer graphics.
3. Write a short note on computer graphics software.
4. Explain various graphics software standards.
5. Explain various graphics functions.
6. List and explain any five input devices used in graphics.

1.7 REFERENCES

- D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Edition, 1997.

1.8 BIBLIOGRAPHY

- J. Hughes, A. Van Dam, M. MCGuire, D. Sklar, J. Foley, and K. Akeley, *Computer Graphics, Principles and Practice*, 3rd Edition, Pearson, 2014.
- P. Shirley and S. Marschner, “Fundamentals of Computer Graphics: Ray Tracing,” *CRC Press*. p. 752, 2009.

DISPLAY TECHNOLOGIES

Unit Structure

2.0 Objectives

2.1 Display Devices

2.1.1 Storage Tubes Graphics Displays

2.1.2 Calligraphic Refresh Graphics Displays

2.1.3 Raster Refresh Graphics Displays

2.1.4 Random Scan Display v/s Raster Scan Display

2.1.5 Cathode Ray Tube (CRT)

2.1.6 Color CRT

2.1.6.1 Beam-Penetration Technique

2.1.6.2 Shadow Mask Technique

2.1.7 Liquid Crystal Display (LCD)

2.1.8 LCD Displays v/s CRT Monitors

2.2 Raster Scan Systems

2.2.1 Video Controller

2.2.2 Display Processor

2.3 Random Scan Systems

2.4 Summary

2.5 Unit End Exercise

2.6 References

2.7 Bibliography

2.0 OBJECTIVES

After this chapter, student will be able to:

- Identify various display devices.
- State the working of display devices.
- State and differentiate raster scan and random scan devices.

2.1 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.

2.1.1 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:

- Two Electron Guns
- Phosphor coated screen,
- Storage grid mesh
- Collector

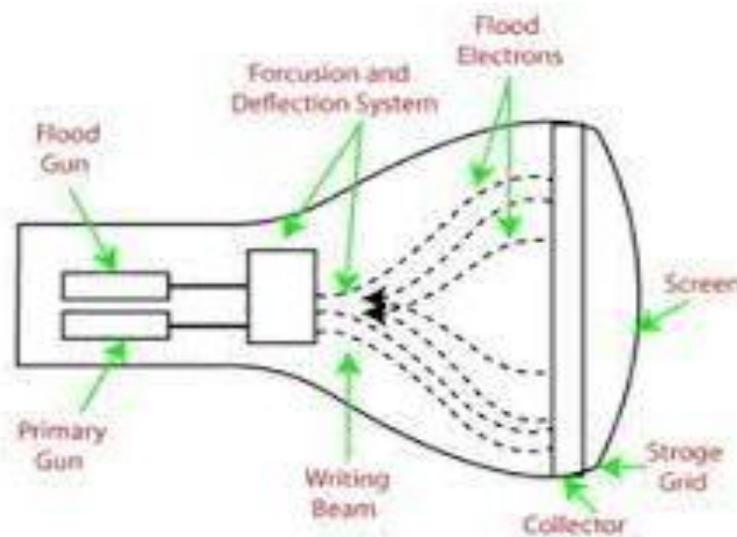


Fig 1. Working of Direct View Storage Tube

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.

2.1.2 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 directed only to the parts of the screen where a picture is to be drawn.
- Random scan monitors draw a picture one line at a time.
- 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

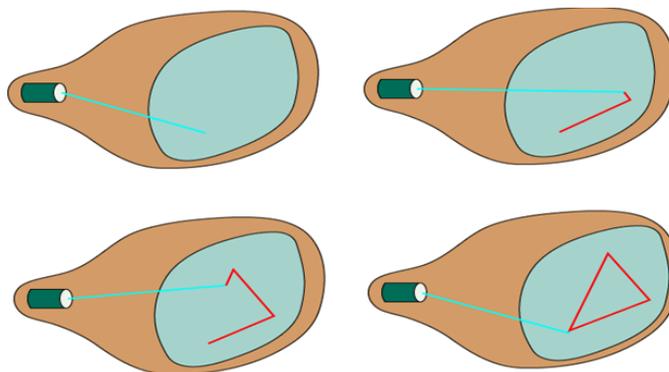


Fig 2. Working of Random Scan Systems

- Random scan displays draw all the component lines of a picture with 30 to 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; hence it 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.

2.1.3 Raster Refresh Graphics Displays:

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

Working Principle:

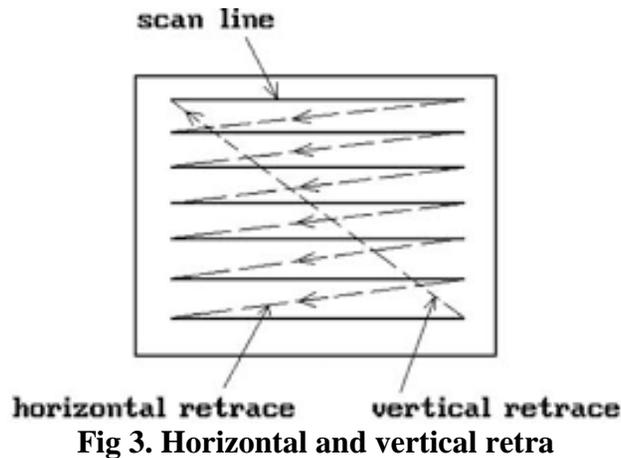
- In this method, horizontal and vertical deflection signals are generated to move a beam all over the screen in a pattern for displaying any image.
- The electron beam is swept across the screen one row at a time from top to bottom.
- The electron beam sweeps back and forth from left to right across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



- 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.

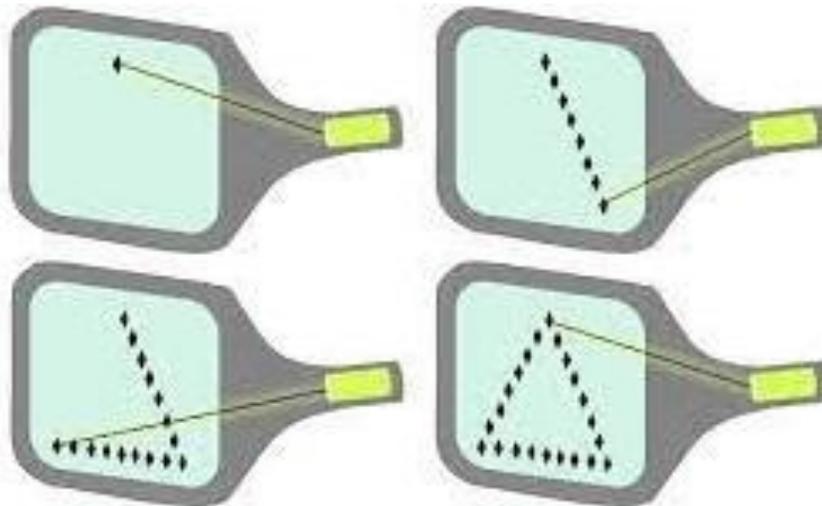


Fig 4. Working of Raster Scan Systems

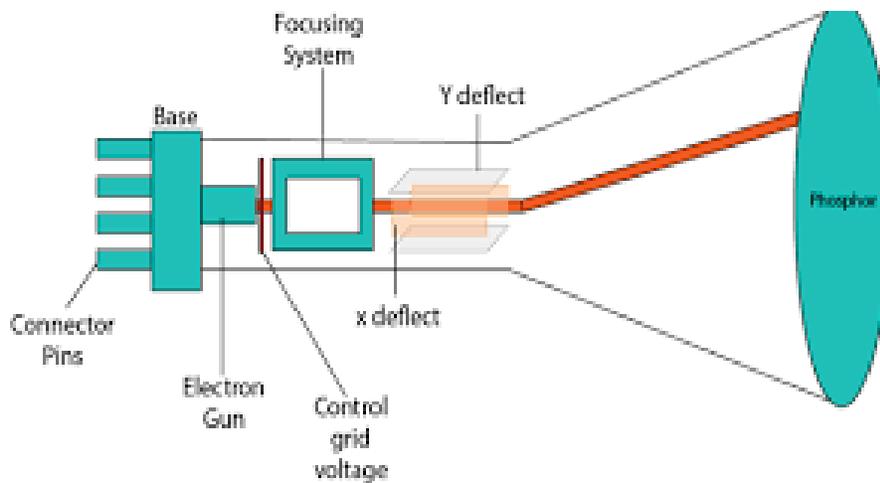
2.1.4 Random Scan Display V/S Raster Scan Display:

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.

2.1.5 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 screen, namely Random Scan and Raster Scan.



Cathode Ray Tube

Fig 5. Working of CRT Monitors

2.1.6 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

2.1.6.1 Beam-Penetration Techniques:

- This technique is used with random-scan monitors.
- In this technique, the inside of CRT screen is coated with two layers of phosphor, usually red and green.
- The displayed color depends on how far the electron beam penetrates into the phosphor layers.
- The outer layer is of red phosphor and the inner layer is of green phosphor.
- A beam of slow electrons excites only the outer red color.
- A beam of very fast electrons penetrates through the red layer and excites the inner green layer.
- At intermediate beam speeds, combination of red and green light are emitted 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.

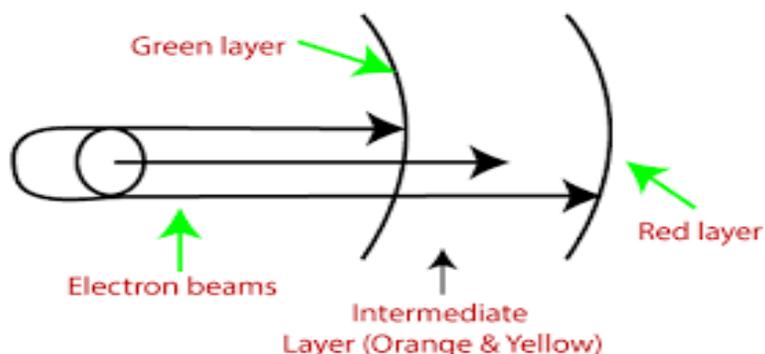


Fig 6. Beam Penetration Technique

2.1.6.2 Shadow Mask Technique:

- The shadow mask technique produces a much wider range of colors than 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 at each pixel position.
- One phosphor dot emits a red light, second emits green light and third emits 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.
- Three electron beams are deflected and focused as a group on to the shadow mask and when they pass through a hole in the shadow mask, they excite a dot triangle.
- A dot triangle consists of three small phosphor dots of red, green and blue color.
- A dot triangle when activated appears as a small dot on the screen, which has color of combination of three small dots.
- By varying the intensity of three electron beams, we can obtain different colors in the shadow mask CRT.

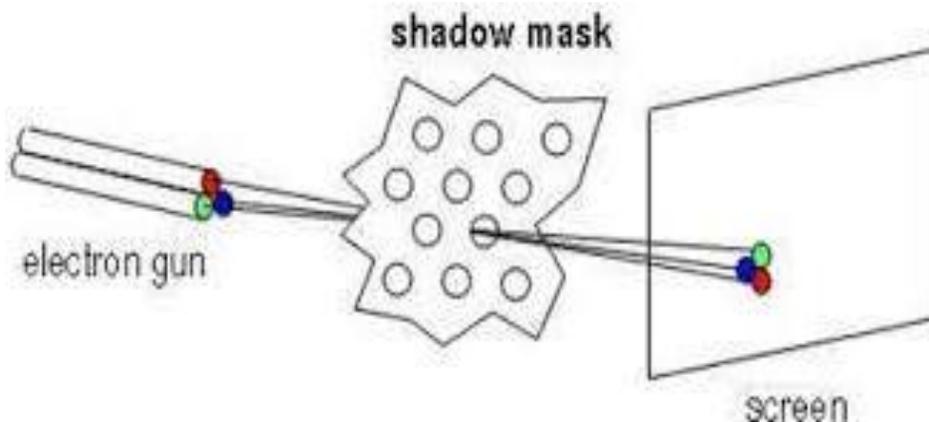


Fig 7. Shadow Mask Technique

2.1.7 Liquid Crystal Displays (Lcd):

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

- The intersection of two conductors defines a pixel position.
- In the ON state, polarized light passing through material is twisted so that 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 intersecting conductors 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.

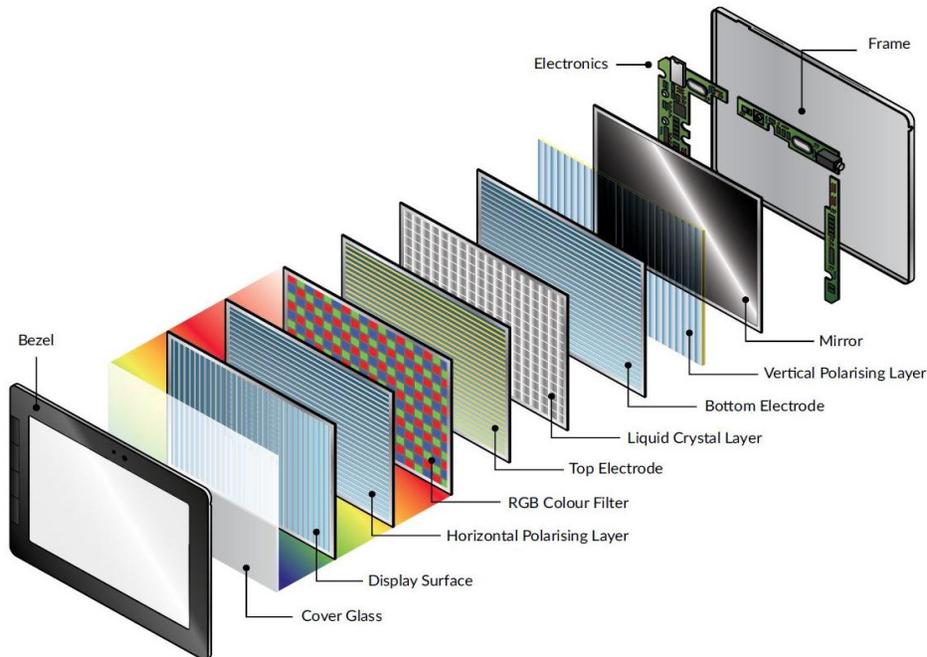


Fig 8. Working of LCD

2.1.8 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.

2.2 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.

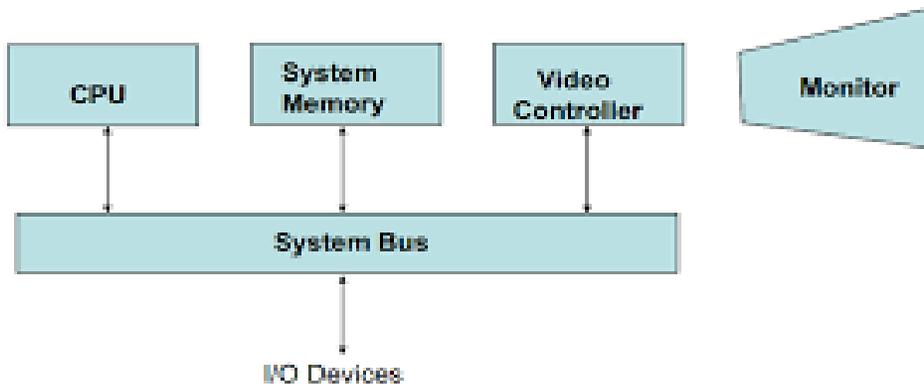


Fig 9. Architecture of a Raster Scan System

- Video controller reads the command from frame buffer and draw pixel accordingly.

2.2.1 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.
- X coordinate increases from left to right and Y coordinate increases from bottom to top.

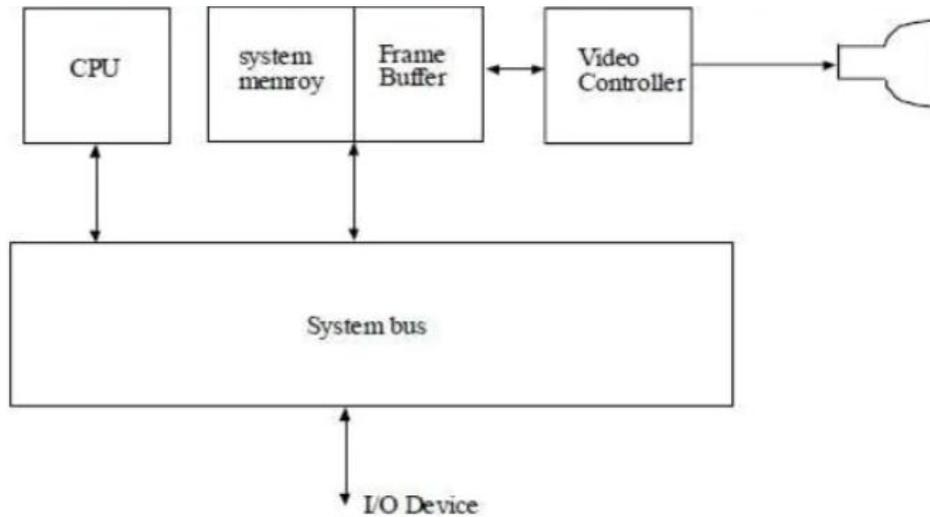


Fig 10. Architecture of a Raster Scan System

- In some systems, origin is set at the top left corner. In such case, Y coordinate increases from top to bottom.
- Two register are maintained to store the coordinates of pixel on screen.
- Intensity value from frame buffer is retrieved from current (x, y) location.
- After painting each pixel, x register is incremented by 1 and process is repeated until x value hit to x_{max} .
- After processing all the pixels on current scan line, content of x register is set to 0, and y is incremented by 1, and video controller processes the same way.
- Refresh rate is 60 frames/second.
- This is shown in Fig 11.
- In high quality systems, use two frame buffers, so that one buffer can be used for refreshing while other is being filled with intensity values.
- Then both buffer switch the role.
- This mechanism is used in real time animation because it does not waste time in reloading the buffer.

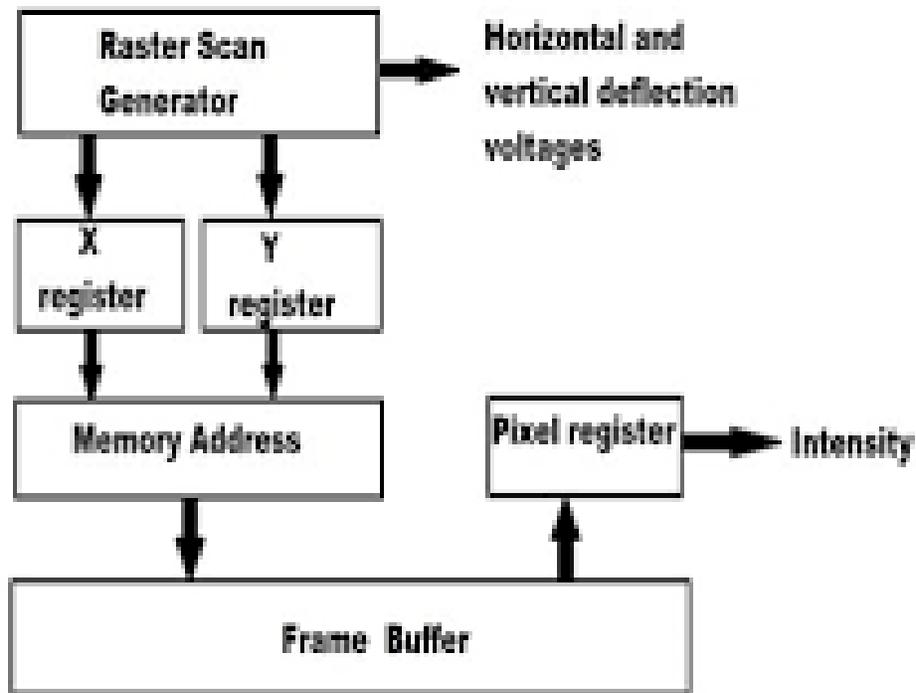


Fig 11. Refresh operation on video controller

2.2.2 Display Processor :

- Display processor is also known as graphics controller or a display coprocessor.
- Its purpose is to free the CPU from graphics operations.
- For this, a separate display processor memory area is provided as shown in Fig 12.
- 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 objects are 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 position in the frame buffer.
- This method can also be used to scan convert curved lines or polygon outlines.
- Display processor can also be used to generate various line styles, displaying color areas, performing transformations and manipulating objects.

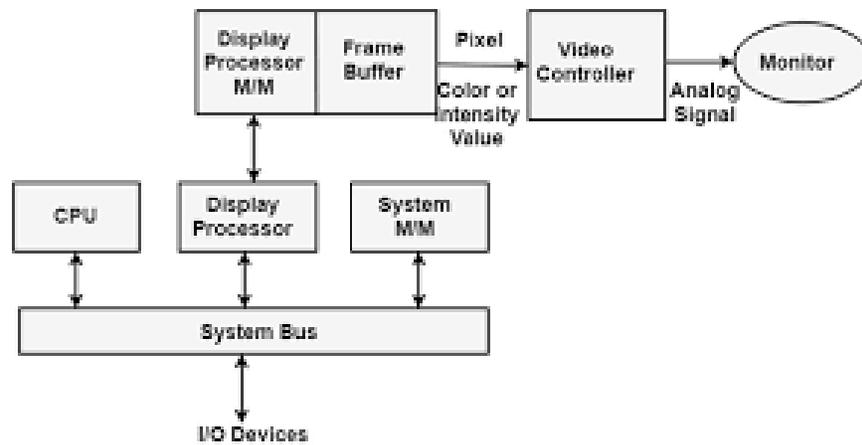


Fig 12. Raster Scan System with a display processor

2.3 RANDOM SCAN SYSTEMS

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

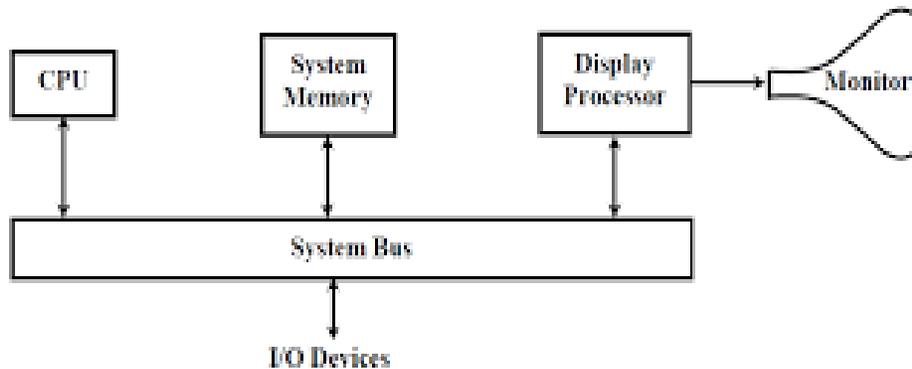


Fig 13. Random Scan System

- Application program resides in system memory.
- Graphics packages translate the graphics commands in the application program into a display file.
- Display file is also stored in system memory.
- Display processor accesses the content of display file and display on monitor screen.
- Display processor retrieves one by one command from display file and draw 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 through each and every pixel on screen.
- Entire scene is defined using segment of line in in application program.

- Line segments are defined by pair of end points.
- A scene is then drawn one line at a time by positioning the beam to fill the line between specified points.

2.4 SUMMARY

- The display devices are the means of output devices.
- Random scan displays stores picture definition as a set of line commands.
- Raster scan displays plots image pixel by pixel on the screen.
- The main components of CRT are electron gun, control grid, focusing system, horizontal and vertical deflection plates.
- Color CRT uses two techniques: beam penetration and shadow mask.
- In beam penetration technique, two colors red and green are used which generates orange and yellow. So only four colors are produced.
- In shadow mask technique, the three-phosphor dots produced are red, green and blue, which produces more number of colors.
- LCD are portable that consumes less electricity.

2.5 UNIT END EXERCISE

1. Differentiate between raster scan and random scan displays.
2. Explain the working of CRT.
3. Explain the shadow mask technique.
4. Explain the raster scan system.
5. Explain random scan systems.
6. Differentiate between LCD and CRT.

2.6 REFERENCES

- D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Edition, 1997.

2.7 BIBLIOGRAPHY

- J. Hughes, A. Van Dam, M. MCGuire, D. Sklar, J. Foley, and K. Akeley, *Computer Graphics, Principles and Practice*, 3rd Edition, Pearson, 2014.
- P. Shirley and S. Marschner, “Fundamentals of Computer Graphics: Ray Tracing,” *CRC Press*. p. 752, 2009.

SCAN CONVERSION – LINE AND CIRCLE

Unit Structure

- 3.0 Objectives
- 3.1 Scan Conversion of Points
- 3.2 Scan Conversion of Lines
 - 3.2.1 DDA Line Drawing Algorithm
 - 3.2.2 Example of DDA Line Drawing Algorithm
 - 3.2.3 Bresenham's Line Drawing Algorithm
 - 3.2.4 Example of Bresenham's Line Drawing Algorithm
- 3.3 Scan Conversion of Circle
 - 3.3.1 Midpoint Circle Algorithm
 - 3.3.2 Bresenham's Circle Algorithm
 - 3.3.3 Midpoint Ellipse Algorithm
- 3.4 Aliasing and Problems in aliasing
- 3.5 Summary
- 3.6 Unit End Exercise
- 3.7 References
- 3.8 Bibliography

3.0 OBJECTIVES

After studying this chapter, student will be able to:

- State and explain various line drawing algorithms
- Generate lines based on line algorithms
- State and explain circle and ellipse algorithms
- Generate circle based on algorithms
- Define aliasing and its problems

3.1 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, polygon and other shapes.
- Graphics objects are continuous.
- Pixels are always discrete. Each pixel has two states: on and off. 0 is addressed by pixel off and 1 is addressed by pixel on.
- The video display device of the computer has a circuit that converts the binary 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.

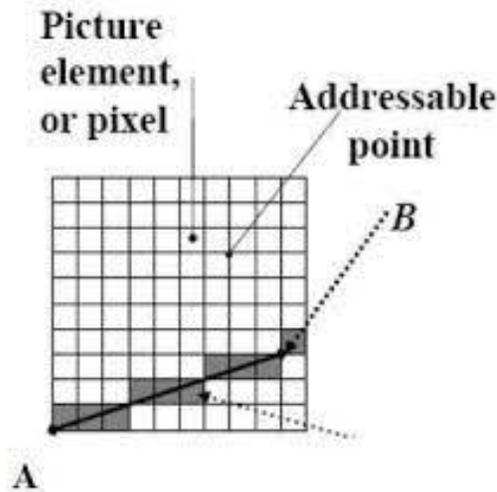


Fig 1. Pixel representation

3.2 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.

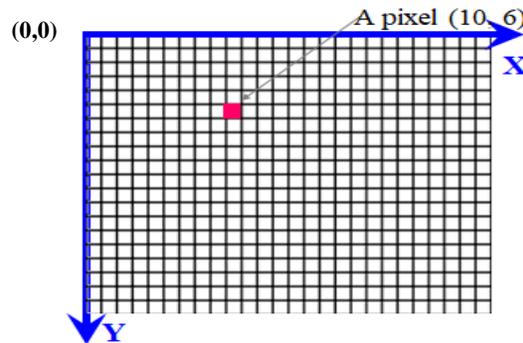


Fig 2. A pixel representation as screen coordinates on a display device

- The values in the x-coordinate increases from left to right and the values of the y-coordinate increases from top to bottom.
- A mathematical point (x, y) where x and y are real numbers within an image area, needs to be scan-converted to a pixel location at (x', y') .
- Each pixel on graphic display does not represent a mathematical point like $P(2.9, 4.22)$.
- 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 equal to the argument

$$x' = \text{Floor}(x)$$

$$y' = \text{Floor}(y)$$

for point $P(2.9, 4.22)$.

$$x' = \text{Floor}(2.9) \text{ and } y' = \text{Floor}(4.22),$$

$$(x', y') \text{ will be } (3, 4).$$

3.3 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 $P2$ as shown in Fig 3.

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.

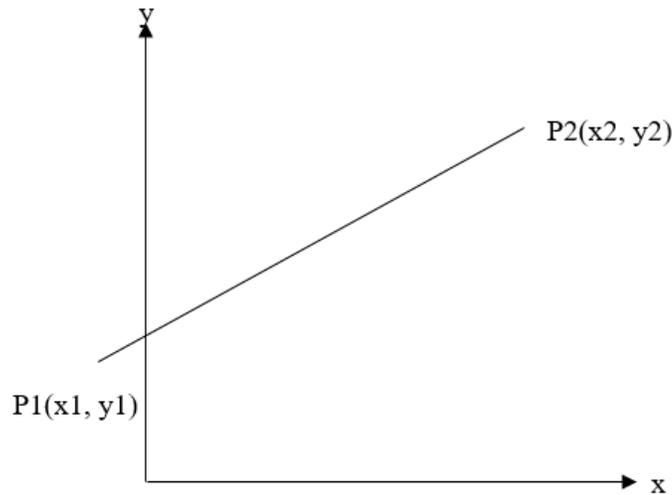


Fig 3. Line representation on coordinate system

3.3.1 DDA Line Drawing Algorithm

DDA stands for Digital Differential Analyzer. This algorithm works on incremental approach.

Drawing Concept

- 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 the end 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.

$$\Delta y = (y_2 - y_1)$$

$$\Delta x = (x_2 - x_1)$$

where,

(x_1, y_1) is the start point

(x_2, y_2) is the end point

Δy is the difference between y coordinates

Δx is the difference between x coordinates

- The slope is calculated as:

$$m = \frac{\Delta y}{\Delta x} \Rightarrow \Delta y = m\Delta x$$

- The DDA algorithm is based on the values of Δx and Δy .
- 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 move down (along the y-axis) by m pixels (based on screen coordinates).

Algorithm:

1. Start
2. Input the two end points of a line as (x_1, y_1) and (x_2, y_2)
3. Compute
 $dx = x_2 - x_1$ and $dy = y_2 - y_1$
4. if $abs(dx) > abs(dy)$
 $step = abs(dx)$
else
 $step = abs(dy)$
5. Compute
 $xinc = dx/step$
 $yinc = dy/step$
6. Initialize $x = x_1$ and $y = y_1$
7. Plot pixel (x, y)
8. For $k=1$ to $step$
{
 $x = x + xinc$
 $y = y + yinc$
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.

3.3.2 Example of DDA Line Drawing Algorithm:

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

Solution to trace line PQ.

1. Start
2. Input $x_1=0, y_1=0, x_2=7, y_2=4$
3. Compute
$$dx=x_2-x_1=7-0=7$$
$$dy=y_2-y_1=4-0=4$$
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

- Fig 4 shows the screen plot of line PQ.

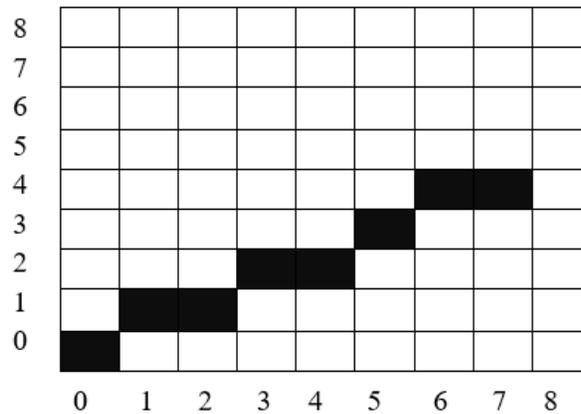


Fig 4. Screen plots of line PQ

3.3.3 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 (x_1, y_1) and (x_2, y_2)
3. Plot pixel (x_1, y_1)
4. Compute

$$dx = x_2 - x_1$$

$$dy = y_2 - y_1$$
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 (x_2, y_2)
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.

3.3.4 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 $x_1=0, y_1=0, x_2=7, y_2=4$
3. Plot (0, 0)
4. Compute

$$dx=x_2-x_1=7-0=7$$

$$dy=y_2-y_1=4-0=4$$

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 parameter p_k

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})

compute $p_{k+1} = p_k + 2*dy - 2*dx$

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

- Fig 5 shows the screen plot of line AB.

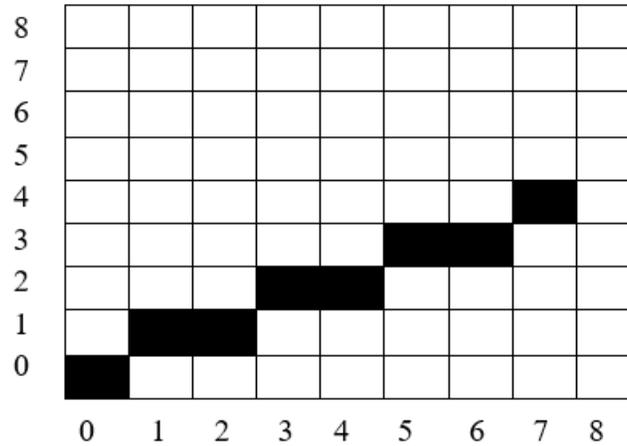


Fig 5. Screen plots of line AB

3.4 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.

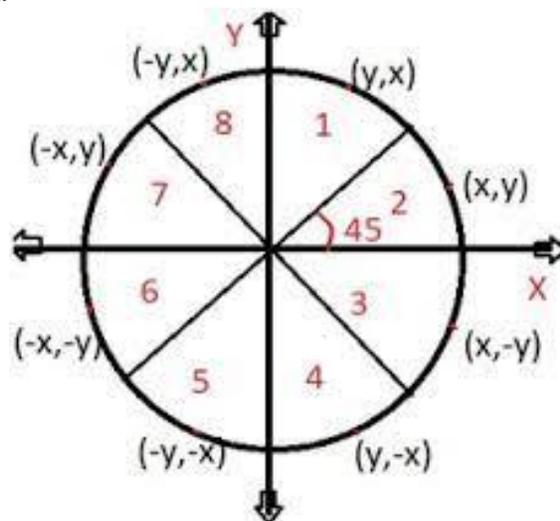


Fig 6. Midpoint criteria

- 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° 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 coordinates
 r is the radius
- The second approach describe the circle using trigonometric relation, where x and y coordinates of the circle is given by $x=r \cos\theta$ and $y=r \sin\theta$, and θ is the current angle.

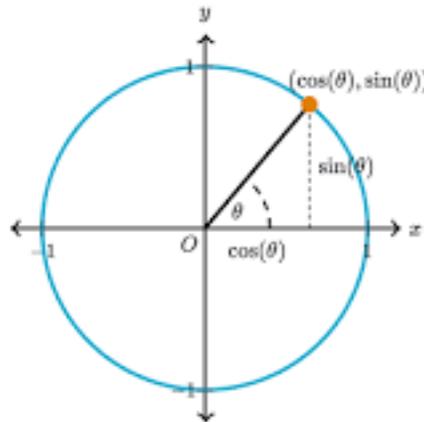


Fig 7. Representing circle in trigonometric relations

3.4.1 Midpoint Circle Algorithm:

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

- The perimeter points will be calculated for first octant and will be printed 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.

3.4.2 Bresenham's Circle Algorithm :

- The Bresenham's circle drawing algorithm calculates all the points nearest 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 is faster.

- 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 $x \leq y$
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.

3.4.3 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 remains constant.

- It has two axis, namely major axis and minor axis, where the major axis is the longest diameter and minor axis is the shortest diameter.
- Ellipse has a four-way symmetry property, which means quadrants are symmetric and not the octants.
- Midpoint ellipse algorithm plots the point of an ellipse on the first quadrant 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.

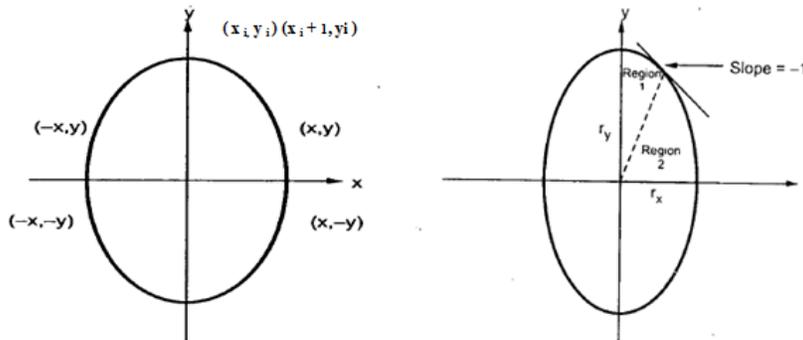


Fig 8. Four-way symmetry and Criteria for Ellipse

- Consider the equation of ellipse as $f(x, y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$ 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 three conditions 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, P1 for 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, $P2 = r_y^2 (x + 1 / 2)^2 + r_x^2 (y - 1)^2 - r_x^2 r_y^2$

• **Algorithm:**

1. Start
2. Declare r_x , r_y , x , y , dx , dy , $P1$, $P2$ (r_x is x radius, r_y = y radius, x = x coordinate to plotted , y = y coordinate to be plotted, dx , dy , $P1$ = decision parameter for region 1, $P2$ = decision parameter for region 2)
3. Initialize initial point for region 1 as $x = 0$, $y = r_y$
4. Calculate $P1 = r_y^2 + r_x^2 / 4 - r_y r_x^2$, $dx = 2 r_y^2 x$, $dy = 2 r_x^2 y$
5. Update values of dx and dy after every iteration.
6. Repeat while ($dx < dy$)
Plot (x , y)
If ($P < 0$)
 $x = x + 1$
 $P1 = P1 + r_y^2 (2*x + 3)$
Else
 $x = x + 1$
 $y = y - 1$
7. When $dx \geq dy$, plot region 2.
8. Calculate $P2 = r_y^2(x + 1 / 2)^2 + r_x^2(y - 1)^2 - r_x^2 r_y^2$
9. Repeat till ($y > 0$)
If ($P2 > 0$)
 $y = y - 1$
 $P2 = P2 - 2*y*r_x^2 + r_x^2$
Else
 $x = x + 1$
 $y = y - 1$
 $P2 = P2 + 2*r_y^2*(2*x) - 2*y*r_x^2 + r_x^2$
10. End

Merits:

- It is easy to implement.

Demerits:

- It is a time consuming algorithm.

3.5 ALIASING AND PROBLEMS IN ALIASING

- In computer graphics, smooth curves and other lines become 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.

3.6 SUMMARY

- Scan conversion is an essential method for plotting discrete pixels.
- DDA line drawing algorithm uses floating-point values, which is converted to integer while plotting pixel. Therefore, it is time consuming.
- Bresenham's line drawing algorithm is more efficient and faster than DDA, as it involves only integer values.
- Midpoint circle algorithm is efficient and easy to implement and works best for raster displays.
- Bresenham's circle algorithm is simple to implement but does not work on complex images.
- Midpoint ellipse algorithm works on criteria of two regions and is time consuming.
- Aliasing occurs on low resolution screens, where smooth curves or lines become jagged.

3.7 UNIT END EXERCISE

1. Explain the DDA line drawing algorithm.
2. Explain the Bresenham's line drawing algorithm with example.
3. Given the endpoints of line XY as (2, 3) and (8, 9). Apply DDA line drawing algorithm on line XY and plot the pixels.

4. Explain the midpoint circle algorithm.
5. Write the Bresenham's circle algorithm.
6. Write the algorithm for midpoint ellipse.
7. Write a short note on aliasing.

3.8 REFERENCES

- D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Edition, 1997.
- R. K. Maurya, *Computer Graphics*, John Wiley.

3.9 BIBLIOGRAPHY

- J. Hughes, A. Van Dam, M. MCGuire, D. Sklar, J. Foley, and K. Akeley, *Computer Graphics, Principles and Practice*, 3rd Edition, Pearson, 2014.
- P. Shirley and S. Marschner, "Fundamentals of Computer Graphics: Ray Tracing," *CRC Press*. p. 752, 2009.

SCAN CONVERSION – LINE AND POLYGON CLIPPING

Unit Structure

- 4.0 Objectives
- 4.1 What is clipping?
- 4.2 Point Clipping
- 4.3 Line Clipping
 - 4.3.1 Cyrus-Beck Line Clipping Algorithm
 - 4.3.2 Cohen-Sutherland Line Clipping Algorithm
 - 4.3.3 Liang-Barsky Line Clipping Algorithm
- 4.4 Polygon Clipping
 - 4.4.1 Sutherland-Hodgeman Polygon Clipping
- 4.5 Summary
- 4.6 Unit End Exercise
- 4.7 References
- 4.8 Bibliography

4.0 OBJECTIVES

After studying this chapter, student will be able to:

- Identify clipping region with region codes
- Implement various clipping algorithms on lines and polygon
- Extract scene with different clipping methods

4.1 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.

- The region against which an object is to be clipped is called a clip window as shown in Fig 1.

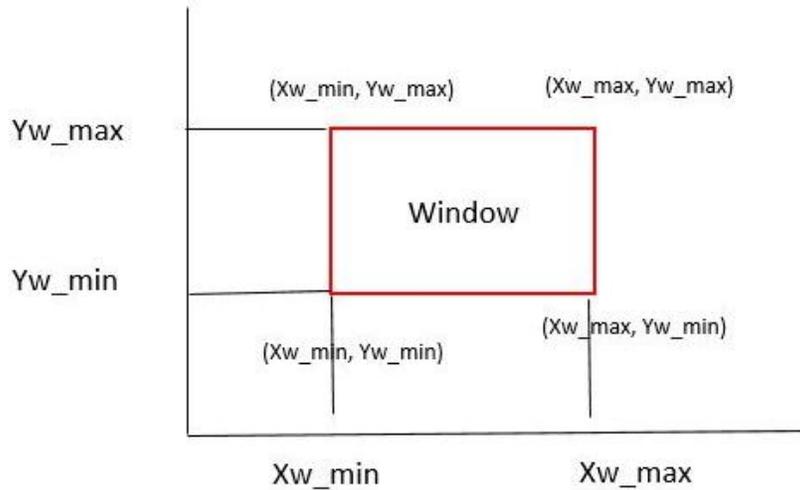


Fig 1. Clipping window

- 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 parts that 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 Fig 2, 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).

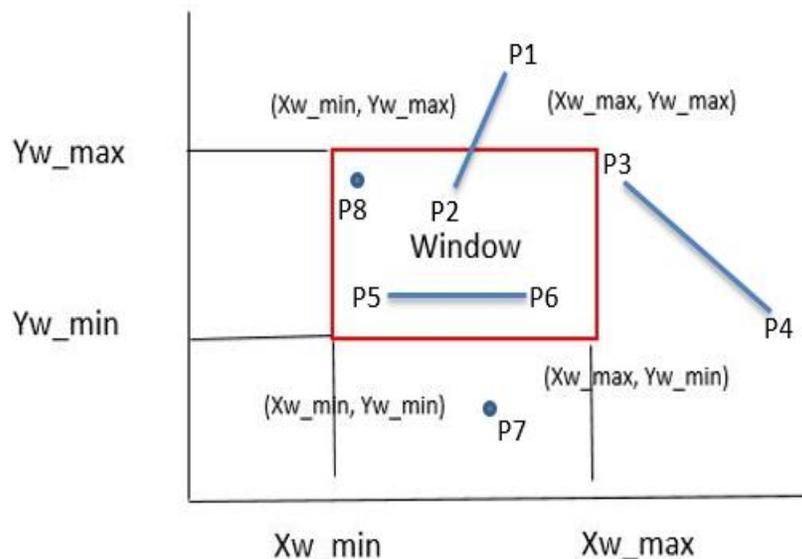


Fig 2. Example of clipping

4.2 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, a point $P(x, y)$ is used for display with the following constraints

$$xW_{\min} \leq x \leq xW_{\max} \text{ AND } yW_{\min} \leq y \leq yW_{\max}$$

where edges of the clip window are xW_{\min} , xW_{\max} , yW_{\min} and yW_{\max} then the point is clipped.

- In Fig 3, 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 D and point E lie outside the window so they are clipped.

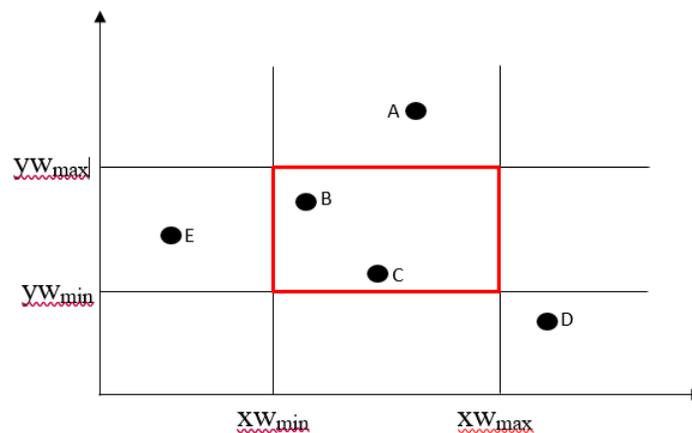


Fig 3. Example of Point Clipping

4.3 LINE CLIPPING

- Removing the portion of the lines which are lying outside the edges of the window 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.

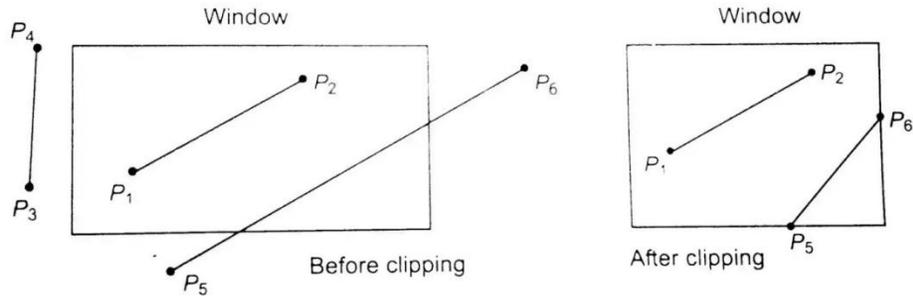


Fig 4. Example of Line Clipping

Examples

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

4.3.1 Cyrus-Beck Line Clipping Algorithm:

Cyrus Beck Line clipping algorithm is a parametric line-clipping algorithm. The term parametric means it is required to find the value of the parameter 't' in the parametric representation of the line segment for the point at that the segment intersects the clipping edge.

- In Fig 5, XY is the line segment, which is intersecting at the two edges A 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 changes the value

- Therefore,

$$X + t(Y-X) = (x_1, y_1) + t((x_2 - x_1), (y_2 - y_1))$$

= (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.

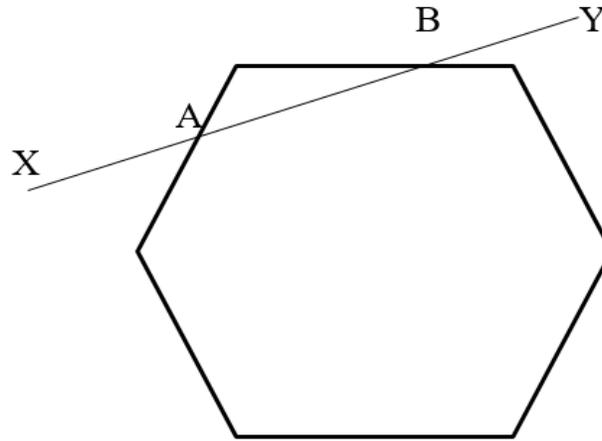


Fig 5. Interaction of line XY with convex polygon window

Merits:

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

Demerits:

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

4.3.2 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.
- Fig 6, defines the clipping regions for Cohen-Sutherland algorithm.

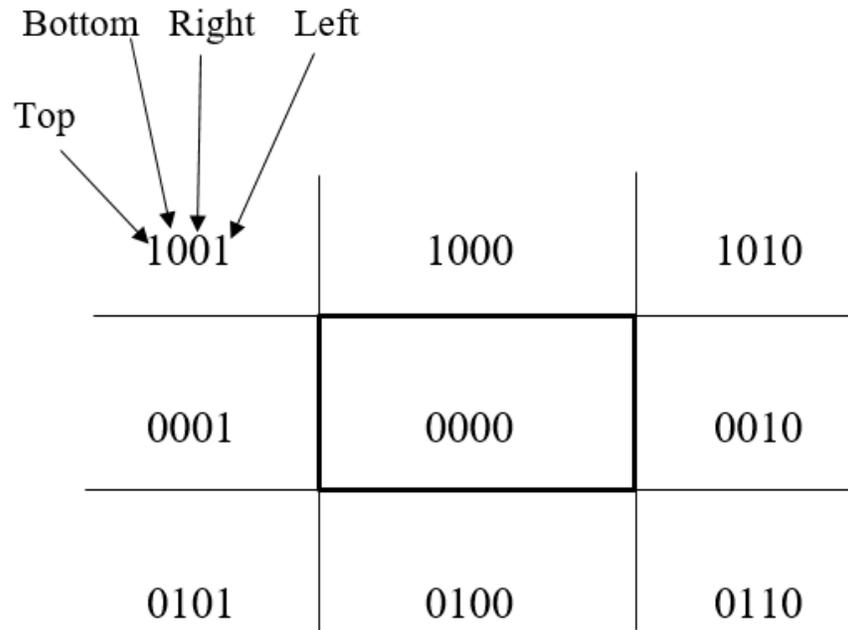


Fig 6 lipping regions

- It uses 4-bits to divide the entire region, and the centre region is the clipping window.
- These 4 bits represents the TOP, BOTTOM, RIGHT and LEFT of the region.
- In the Fig 6, 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 the other convex shaped window.
- Less efficient.

4.3.3 Liang-Barsky Line Clipping Algorithm :

This algorithm is more efficient than Cohen-Sutherland line clipping algorithm 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:

$$\begin{aligned} p_1 &= -dx, & q_1 &= x1 - xw_{min} \\ p_2 &= dx, & q_2 &= xw_{max} - x1 \end{aligned}$$

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

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

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 the line 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 ($t_1 < t_2$)

{

$$\text{newx1} = x1 + t1 * dx$$

$$\text{newx2} = x1 + t2 * dx$$

$$\text{newy1} = y1 + t1 * dy$$

$$\text{newy2} = y1 + t2 * dy$$

Draw line (newx1 , newx2 , newy1 , newy2)

}

11. End

Merits:

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

4.4 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° .

(ii) Convex polygon:

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

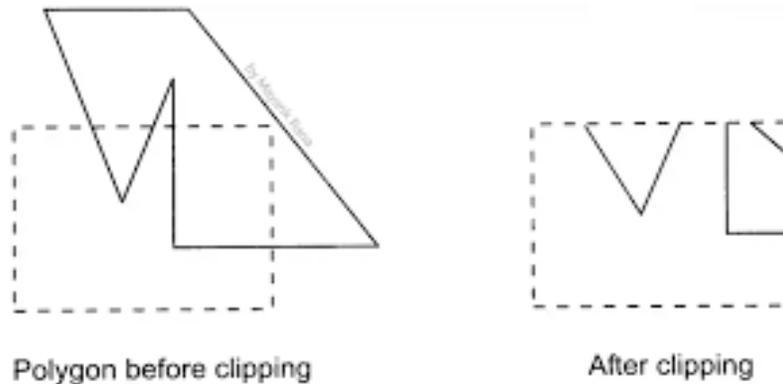


Fig 7. Polygon Clipping

4.4.1 Sutherland-Hodgeman Polygon 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.

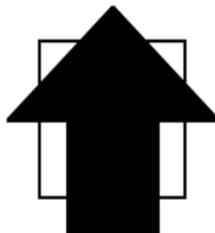


Fig 8. Before Clipping

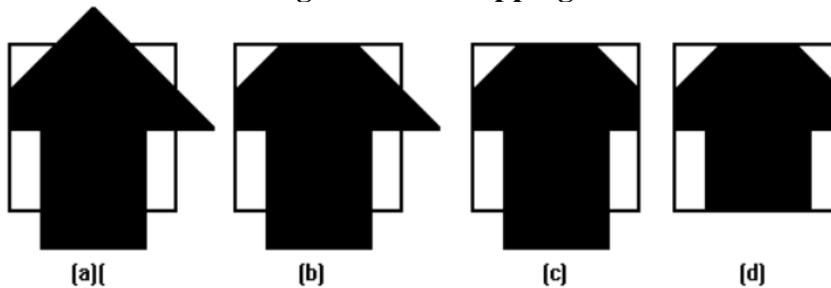


Fig 9. After clipping using Sutherland-Hodgeman algorithm

Example:

- Fig. 9 depicts the following cases:
 - 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 $AV \times AB$ 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 with respect 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.

4.5 SUMMARY

- Cohen-Sutherland is easy to implement line-clipping algorithm, but it is time consuming.
- Cyrus-Beck and Liang-Barsky uses parametric equation for clipping the line.
- Liang-Barsky is faster and most efficient line clipping algorithm.
- Sutherland-Hodgeman is the most efficient algorithm for polygon clipping. However, it requires more memory because new vertices are obtained after clipping against each edge and it requires storing old and new vertices.

4.6 UNIT END EXERCISE

1. Explain the Cohen-Sutherland line-clipping algorithm.
2. Explain the process of point clipping.
3. Write the algorithm for Sutherland-Hodgeman polygon clipping.
4. Explain the process for Cyrus-Beck line-clipping algorithm.
5. Write the algorithm for Liang-Barsky line-clipping algorithm.

4.7 REFERENCES

- D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Edition, 1997.
- R. K. Maurya, *Computer Graphics*, John Wiley.

4.8 BIBLIOGRAPHY

1. J. Hughes, A. Van Dam, M. MCGuire, D. Sklar, J. Foley, and K. Akeley, *Computer Graphics, Principles and Practice*, 3rd Edition, Pearson, 2014.
2. P. Shirley and S. Marschner, “Fundamentals of Computer Graphics: Ray Tracing,” *CRC Press*. p. 752, 2009.

UNIT II

5

TWO DIMENSIONAL TRANSFORMATIONS

Unit Structure

- 5.0 Objective
- 5.1 Introduction
- 5.2 Transformations and Matrices
 - 5.2.1 Transformation
 - 5.2.2 Matrices
- 5.3 Transformation conventions
 - 5.3.1 2D coordinate system
 - 5.3.2 3D coordinate system
- 5.4 2D transformations
 - 5.4.1 Translation
 - 5.4.2 Rotation
 - 5.4.3 Scaling
 - 5.4.4 Shearing
 - 5.4.5 Reflection
- 5.5 Combined Transformation
- 5.6 Homogeneous Coordinates and Matrix Representation of 2D Transformations
- 5.7 Transformation of Points
- 5.8 Transformation of the Unit square
- 5.9 Solid Body Transformations
- 5.10 Rotation about an Arbitrary Point
- 5.11 Reflection through an Arbitrary Line
- 5.12 The Window-to-Viewport Transformations
- 5.13 Summary
- 5.14 Unit End Exercise
- 5.15 References

5.0 OBJECTIVE

This chapter will able you to understand the following concept:

- Transformations and Matrices with its representation
- Types of transformation 2D and 3D

- Methods of transformation
- Ways of Transformation about an Arbitrary point
- Transformation of homogeneous coordinates
- Window-to-Viewport Transformations.

5.1 INTRODUCTION

To convert the real world object into computer we need to understand the coordinate system. The real world objects are represented in computer with the different angles and different perspective with the help of different coordinate system. The concept of transformation is used to implement this process. Transformation takes place in 2 ways that are 2D transformation and 3D transformation.

In this chapter we will see the 2D coordinate system of converting real life object to computer.

5.2 TRANSFORMATIONS AND MATRICES

5.2.1 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.

This change can be done by

1. Geometric transformation:

In this type of transformation the object change its appearance with the help Geometric properties of the specified points. x

2. Coordinate transformation:

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

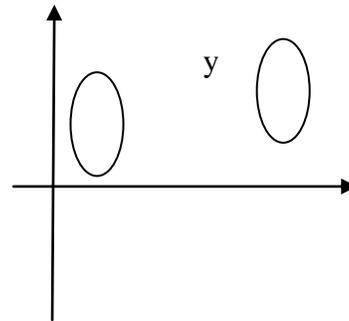


Fig. 5.2.1

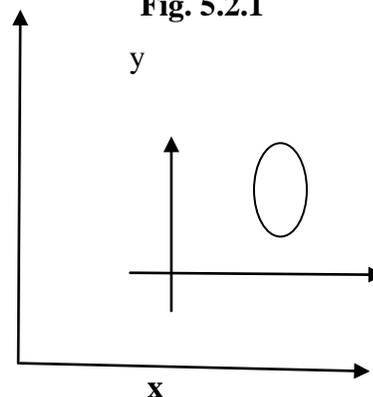


fig. 5.2.2

5.2.2 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 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 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5.3 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.

5.3.1 2D COORDINATESYSTEM:

This 2D coordinate system consists of two coordinates x and y i.e. x plane and 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 on screen with lines, circle, polygon etc

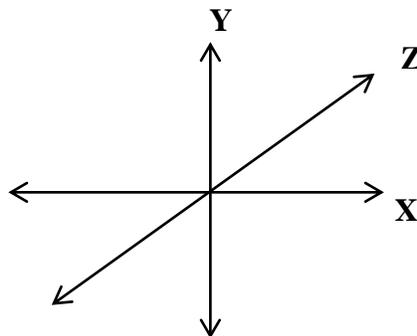


Fig 5.3.1

5.3.2 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.

Rules applied for transformation y

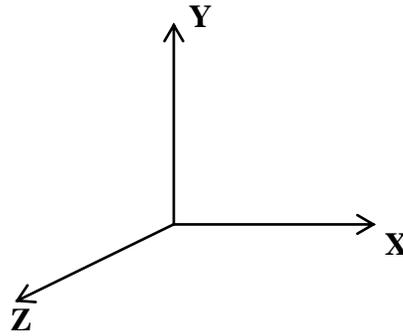


Fig 5.3.2

1. 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 in positive direction only.
origin(0, 0, 0)

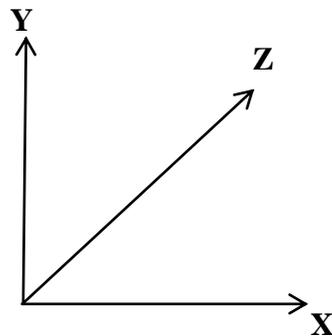


Fig 5.3.3

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.

5.4 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 transformation	2D	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

5.4.1 Translation:

- Changing the position of an object or shifting of an object is called 2D translation.
- 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
- The following matrix shows translation, $P'=P+T$

$$P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} + T \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Example: Apply translation transformation on $\triangle ABC$, with $A=(2,5)$, $B=(2,1)$, $C=(5,1)$ with translation factor $x=2$, $y=2$.

Solution:

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} + T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 3 & 5 & 0 \\ 2 & 2 & 0 \\ 7 & 3 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 2 & 2 \\ 7 & 3 \end{bmatrix}$$

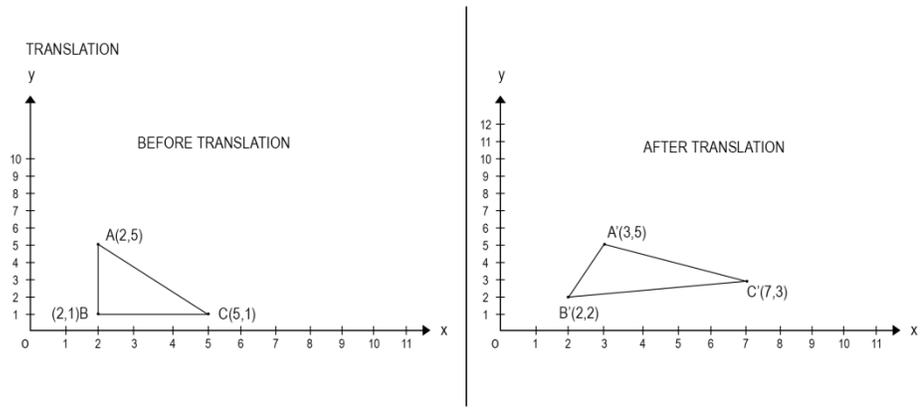


Fig 5.4.1 Translation

5.4.2 Rotation:

- Changing the position of an object with rotation angle along with xy plane is called 2D rotation.
- To achieve rotation an object should move in x and y direction with the specified angle θ .
- We can rotate a two dimensional figure by multiplying **rotation factor r_x and r_y** to the original points x and y
- The following matrix shows rotation, $P'=P+R$

$$\text{Clockwise rotation } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot R \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\text{Anti Clockwise rotation } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot R \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Example: Apply rotation transformation on ΔABC , with $A=(2,5)$, $B=(2,1)$, $C=(5,1)$ with the rotation angle 90° in clockwise direction.

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} \cdot R \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} \cdot R \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = P \begin{bmatrix} -2 & 2 & 0 \\ -1 & 2 & 0 \\ -1 & 5 & 1 \end{bmatrix}$$

by putting $\cos 90^\circ = 0$ and $\sin 90^\circ = 1$

$A'=(-5, 2)$ $B'=(-1, 2)$ $C'=(-1, 5)$

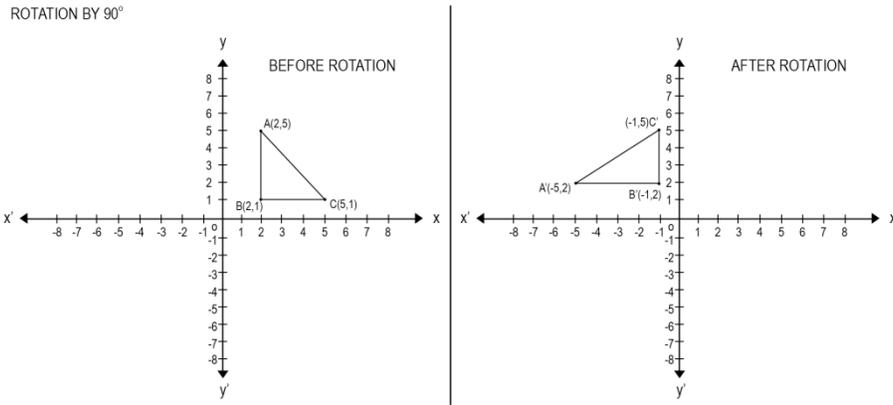


Fig 5.4.2 Rotation

5.4.3 Scaling:

- Changing the size of an object is called 2D scaling.
- To achieve scaling an object should change the size of object in x and y direction.
- We can scale a two dimensional figure by multiplying **scaling factor s_x and s_y** to the original points x and y

The following matrix shows scaling, $P'=P+S$

- Scaling in x and y direction $P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$
- Scaling in x direction

- $P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} s_x & 0 \\ 0 & 1 \end{bmatrix}$

Scaling in y direction $P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} 1 & 0 \\ 0 & s_y \end{bmatrix}$

Example: Apply scaling transformation on ΔABC , with $A=(2,5)$, $B=(2,1)$, $C=(5,1)$ with the scaling factor $x=2$ and $y=2$

Solution: Scaling in x and y direction $P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} \cdot S \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 0 \\ 4 & 2 & 0 \\ 10 & 2 & 1 \end{bmatrix} \text{ by substituting original}$$

value of point A,B,C and scaling factor $x=2$, $y=2$

$A'=(4, 10)$ $B'=(4, 2)$ $C'=(10, 2)$

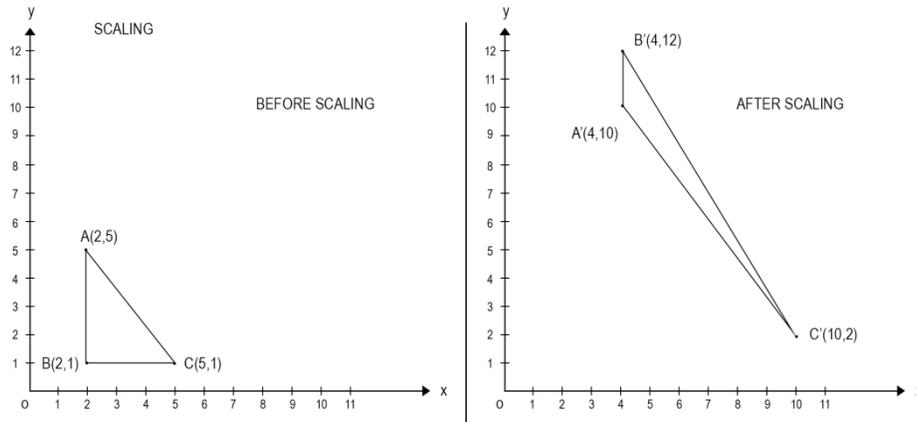


Fig 5.4.3 Scaling

5.4.4 Shearing:

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

The following matrix shows shearing, $P'=P+Sh$

$$P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} \mathbf{0} & \mathbf{shy} \\ \mathbf{shx} & \mathbf{0} \end{bmatrix}$$

$$\text{Shearing with } x \text{ axis } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot Shx \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{shx} & \mathbf{1} \end{bmatrix}$$

$$\text{Shearing with } y \text{ axis } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot Shy \begin{bmatrix} \mathbf{1} & \mathbf{shy} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Example: Apply shearing transformation on $ABC\triangle$ with $A=(2,5)$, $B=(2,1)$, $C=(5,1)$ with the shearing factor $x=2$ and $y=2$ separately.

Solution: Shearing in x direction $P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ Shx & \mathbf{1} \end{bmatrix}$

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} \cdot R \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 5 & 0 \\ 4 & 1 & 0 \\ 6 & 1 & 1 \end{bmatrix} \text{ by substituting}$$

original value of point A,B,C and shearing factor $x=2$

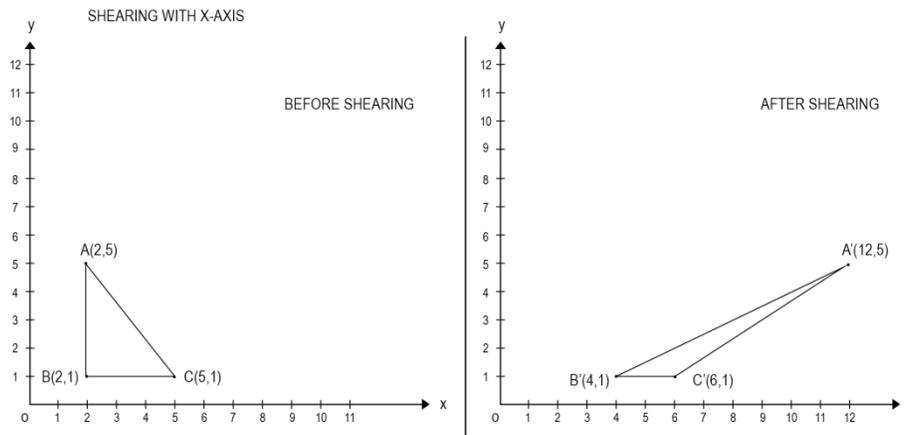


Fig 5.4.4.1 Shearing with X axis

Shearing in y direction $P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot S \begin{bmatrix} 1 & shy \\ 0 & 1 \end{bmatrix}$

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} \cdot R \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 9 & 0 \\ 2 & 5 & 0 \\ 5 & 11 & 1 \end{bmatrix} \text{ by substituting original}$$

value of point A,B,C and shearing factor $y=2$

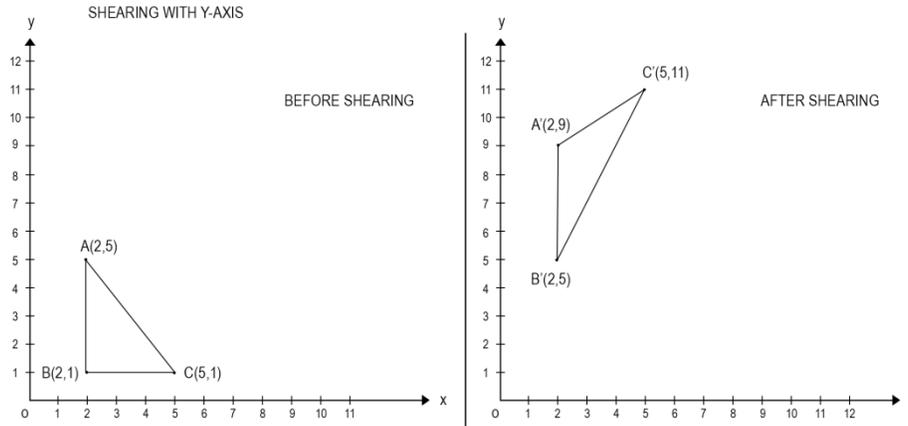


Fig 5.4.4.2 Shearing with Y axis

5.4.5 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 **reflection factor RE_x and RE_y** to the original points x and y

→ The following matrix shows reflection, $P'=P+RE_{xy}$

$$\text{Reflection with } xy \text{ axis } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot RE_x \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\text{Reflection with } x \text{ axis } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot RE_x \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\text{Reflection with } y \text{ axis } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot RE_y \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Example: Apply reflection transformation on ΔABC , with $A=(2,5)$, $B=(2,1)$, $C=(5,1)$

Solution:

$$\text{Reflection with } xy \text{ axis } P' \begin{bmatrix} x' \\ y' \end{bmatrix} = P \begin{bmatrix} x \\ y \end{bmatrix} \cdot RE_x \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P \begin{bmatrix} 2 & 5 & 0 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} \cdot R \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -5 & 0 \\ -2 & -1 & 0 \\ -5 & -1 & 1 \end{bmatrix} \text{ by}$$

substituting original value of point A,B,C and scaling matrix

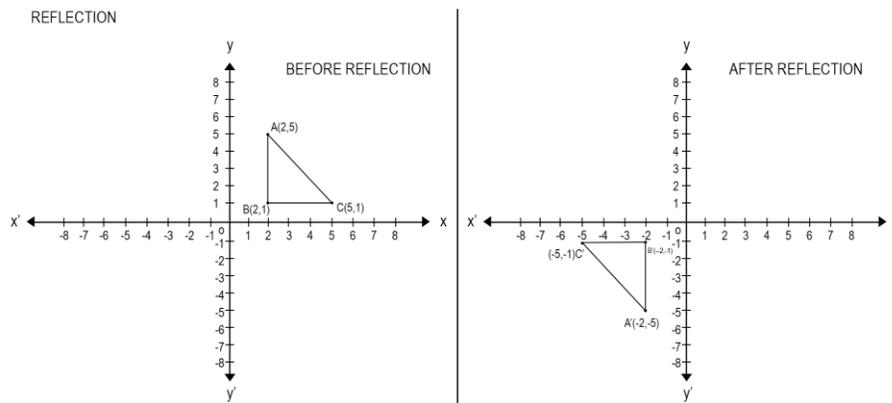


Fig 5.4.5 Reflection

5.5 COMBINED TRANSFORMATION

Some time 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 then we need to apply second transformation on an object in matrix form then the final matrix multiplication is considered for the desired object transformation.

Example: Apply combined transformation Δ on ABC, with A=(2,5), B=(2,1), C=(5,1) apply the following sequence of transformation on ABC

- a) Rotation by angle $\theta=90^\circ$
- b) Reflection through x axis

Solution: The rotation matrix $\theta=90^\circ$ ref fig 5.4.4

$$[R'] = \begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

The reflection through x axis matrix

$$[RE_x'] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The combined transformation matrix is

$$[R'] \cdot [RE_x'] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

Apply combined transformation on original points

$$[P'] = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \cdot ([R'] \cdot [RE_x']) = \begin{bmatrix} 2 & 5 \\ 2 & 1 \\ 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$[P'] = \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \begin{bmatrix} -2 & -5 \\ -2 & -1 \\ -5 & -1 \end{bmatrix}$$

5.6 HOMOGENEOUS COORDINATES AND MATRIX REPRESENTATION OF 2D TRANSFORMATIONS

$$\text{Translation : } \begin{bmatrix} X_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & Tx \\ 0 & 1 & Ty \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

$$\text{Rotation : } \begin{bmatrix} X_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

$$\text{Scaling: } \begin{bmatrix} X \text{ new} \\ y \text{ new} \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \text{ old} \\ Y \text{ old} \\ 1 \end{bmatrix}$$

$$\text{Shearing in x direction: } \begin{bmatrix} X \text{ new} \\ y \text{ new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ SHx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \text{ old} \\ Y \text{ old} \\ 1 \end{bmatrix}$$

$$\text{Shearing in y direction: } \begin{bmatrix} X \text{ new} \\ y \text{ new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & SHy & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \text{ old} \\ Y \text{ old} \\ 1 \end{bmatrix}$$

$$\text{Reflection in x direction: } \begin{bmatrix} X \text{ new} \\ y \text{ new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \text{ old} \\ Y \text{ old} \\ 1 \end{bmatrix}$$

$$\text{Reflection in y direction: } \begin{bmatrix} X \text{ new} \\ y \text{ new} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \text{ old} \\ Y \text{ old} \\ 1 \end{bmatrix}$$

5.7 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 perform transformation on points.

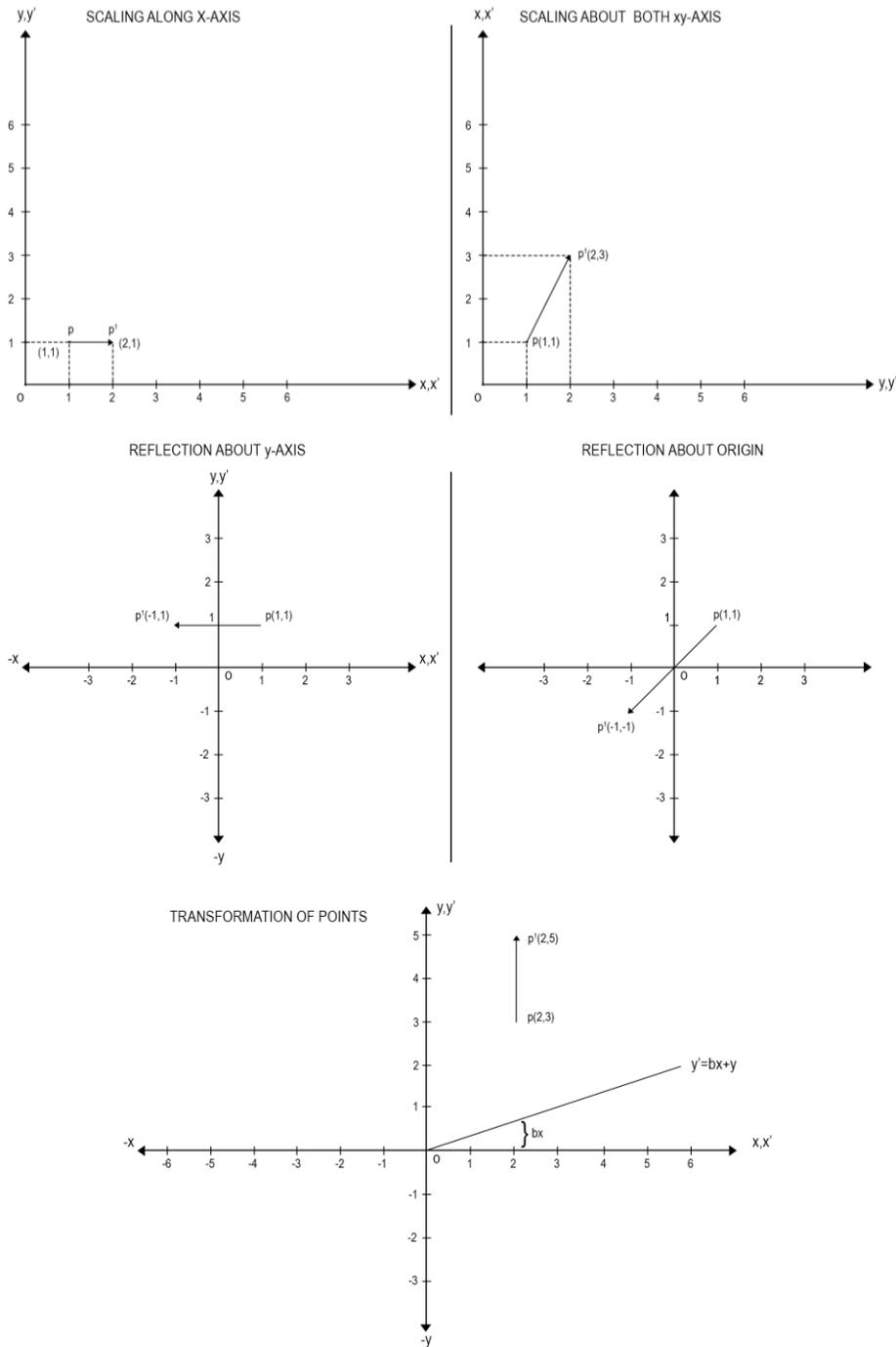


Fig 5.7.1 Transformation of point

5.8 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. As we can see in fig 5.8.1 the origin coordinate remains same i.e. A and A'

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.

The general form of transformation matrix is $[p'] = [p] [T]$

$$[x' \ y'] = [x \ y] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

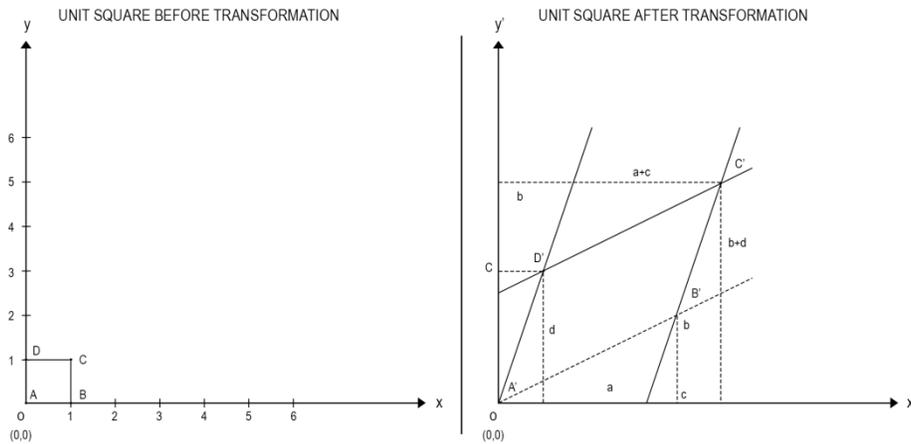


Fig 5.8.1 Unit square transformation

5.9 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 of vector are required.

Vector: A vector is quantity which has direction as well as magnitude. It is 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 another vector which is right angled to both the vector like a x b

5.10 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.

General matrix of homogeneous matrix for translation $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$

When the translation is about origin then m will be changed as -m and n will be changed as -n

Homogeneous matrix for translation about an origin $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix}$

Steps to achieve Rotation about arbitrary points

1. The points of translation about the origin m= -m and n= -n
2. Perform specified rotation.
3. Translate result back to original points (center of origin)
- 4.

$$[x',y',z'] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$$

Example: Center of an object is at point (5,6) and the object is rotated about 90° counter clockwise about its origin using the matrix of rotation.

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Solution: translating the point to the origin i.e m= -m and n= -n;

origin (m,n)m=5 and n=6

$$[x',y',z'] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$$

Origin matrix

rotation

origin translation

$$[x',y',z'] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & -6 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 6 & 1 \end{bmatrix}$$

$$[x',y',z'] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 6 & -5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 6 & 1 \end{bmatrix}$$

$$[x',y',z'] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 11 & 6 & 1 \end{bmatrix}$$

5.11 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 is 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

5.12 THE WINDOW – TO - 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 world 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:

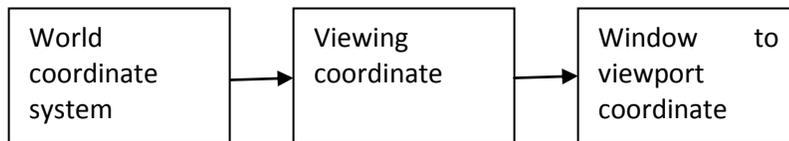


Fig 5.12.1 window to viewport system

Window to viewport mapping:

A point at position (x_w, y_w) in the window is mapped into new position (x_v, y_v) in associated view port.

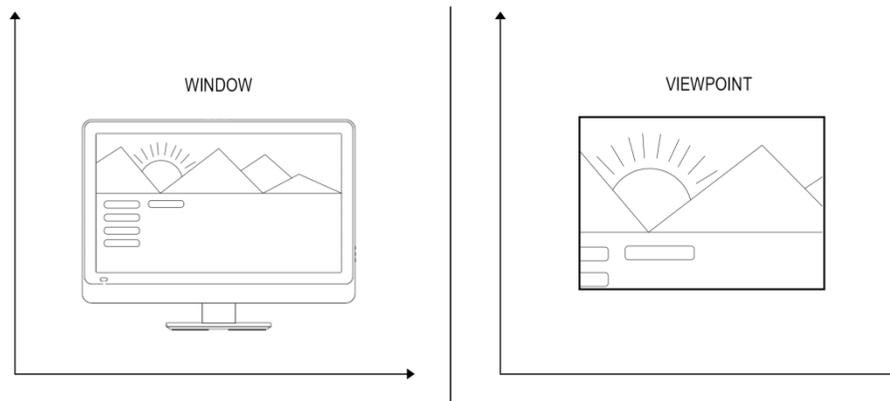


Fig 5.12.2 Window to Viewport transformation

5.13 SUMMARY

The process of changing size, shape, position and orientation of object is called transformation. The basic transformation is translation, rotation, reflection, scaling, shearing. When the object is moved from one position to another position it is called as translation. When we rotate an object in 180° it is called as rotation whereas when we create mirror image of object it is called as reflection. When we change the shape of the object is called as scaling whereas when we create tiled shape of any object it is called as shearing. We can achieve transformation by using multiplication and addition function on points given. Through the transformation we can

change the appearance of the object. we can use 3 X 3 matrixes for 2D transformation whereas 4 X 4 matrixes for 3D transformation. We can get transformation through x and y axis in 2D transformation. We can get transformation through x, y and z axis in 3D transformation. Using more than one transformation we can combined transformation to achieve multiple transformation. We can represent transformations in homogeneous coordinate system.

5.14 UNIT END EXERCISE

1. What is transformation? Explain its types.
2. Explain 2D transformation in detail.
3. Explain homogeneous coordinate system in detail.
4. What is combined transformation? Explain in detail.
5. How to convert an object from window to viewport?
6. Write a short note on rotation about an arbitrary point.
7. Explain in brief solid body transformation.
8. Write a short note on unit square transformation.
9. Explain reflection through an arbitrary line.
10. Explain the concept of transformation of points.

5.15 REFERENCES

- Computer Graphics C version 2nd Edition by Donald D. Hearn and M. Pauline Baker
- Computer Graphics A programming approach 2nd Edition by Steven Harrington McGraw Hill
- Fundamental of Computer Graphics 3rd Edition by Peter Shirley an Steven Marschner

UNIT II

6

THREE DIMENSIONAL TRANSFORMATIONS

Unit Structure

- 6.0 Objective
- 6.1 Introduction
- 6.2 Transformations and Matrices
 - 6.2.1 Transformation
 - 6.2.2 Matrices
- 6.3 Transformation conventions
 - 6.3.1 3D coordinate system
- 6.4 3D transformations
 - 6.4.1 Translation
 - 6.4.2 Rotation
 - 6.4.3 Scaling
 - 6.4.4 Shearing
 - 6.4.5 Reflection
- 6.5 Multiple Transformations
 - 6.5.1 Method
- 6.6 Homogeneous Coordinates and Matrix Representation of 3D Transformations
- 6.7 Rotation about an arbitrary axis in space
- 6.8 Reflection through an arbitrary plane
- 6.9 Affine and Perspective Geometry
- 6.10 Perspective Transformations
- 6.11 Techniques for Generating Perspective Views
- 6.12 Axonometric Projection
- 6.13 Oblique Projection
- 6.14 Orthographic Projection
- 6.15 Vanishing Points
- 6.16 View volumes for projections
- 6.17 Summary
- 6.18 Unit End Exercise
- 6.19 References

6.0 OBJECTIVE

- This chapter will able you to understand the following concept
- 3D Transformations and Matrices with its representation
- Types of transformation such as scaling, reflection, translation, shearing and rotation.
- Ways of Transformation about an Arbitrary plane and arbitrary axis.
- Types of projection such as Orthographic, Axonometric, Oblique Projections.
- Camera model and its perspective geometry

6.1 INTRODUCTION

When we want to convert the real life object into computer we need 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.

6.2 TRANSFORMATIONS AND MATRICES

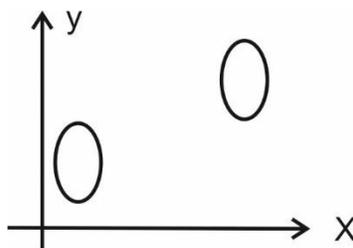
6.2.1 Transformation:

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 is called transformation.

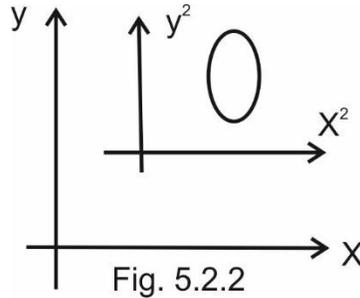
This change can be done by **Geometric transformation:**

In this type of transformation the object changes its appearance with the help 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.



6.2.2 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 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

6.3 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.

6.3.1 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 world object can be drawn on screen with lines, circles, polygons, etc. Rules applied for transformation

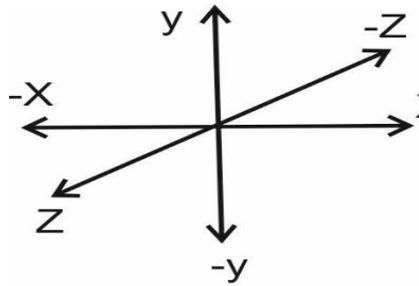


fig 6.3.1

1. 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.

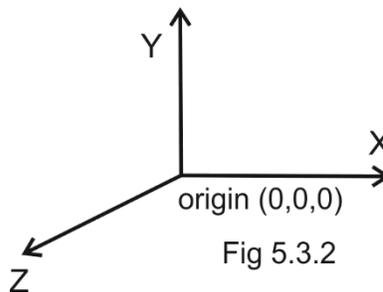


Fig 5.3.2

fig 6.3.2

2. 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.

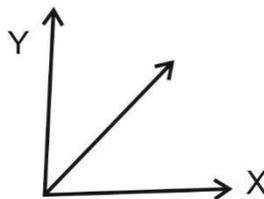


Fig 6.3.3

- 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.

6.4 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

6.4.1 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.
- The following matrix shows translation, $P'=P+T$

$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

General representation of 3D translation matrix

$$P' \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = P [a, b, c, 1] + T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ l & m & n & 1 \end{bmatrix}$$

3-D TRANSLATION

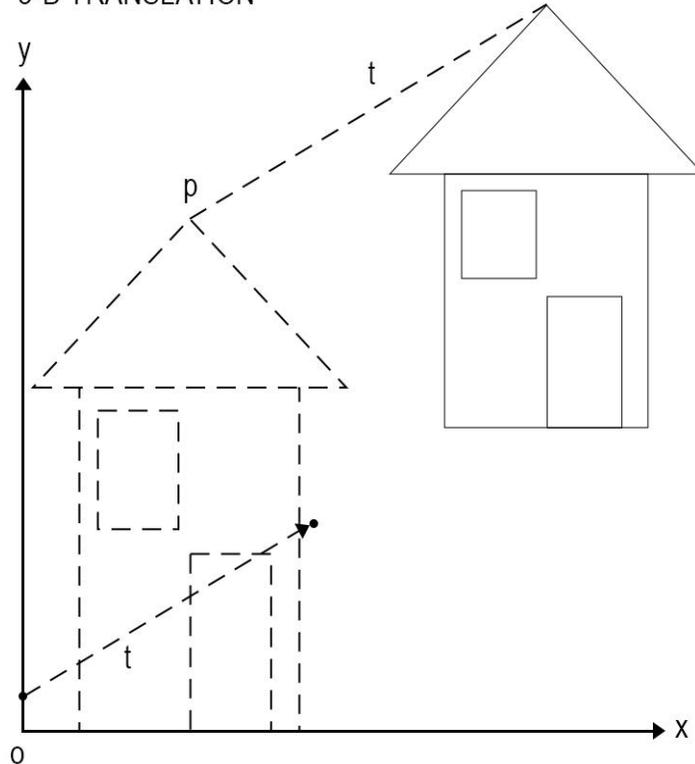


Fig 6.4.1 3D Translation

6.4.2 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 θ .
- 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
- The following matrix shows rotation, $P'=P+R$

$$\text{Rotation about } x \text{ axis } P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot R \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

$$\text{Rotation about } y \text{ axis } P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot R \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\text{Rotation about } z \text{ axis } P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot R \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

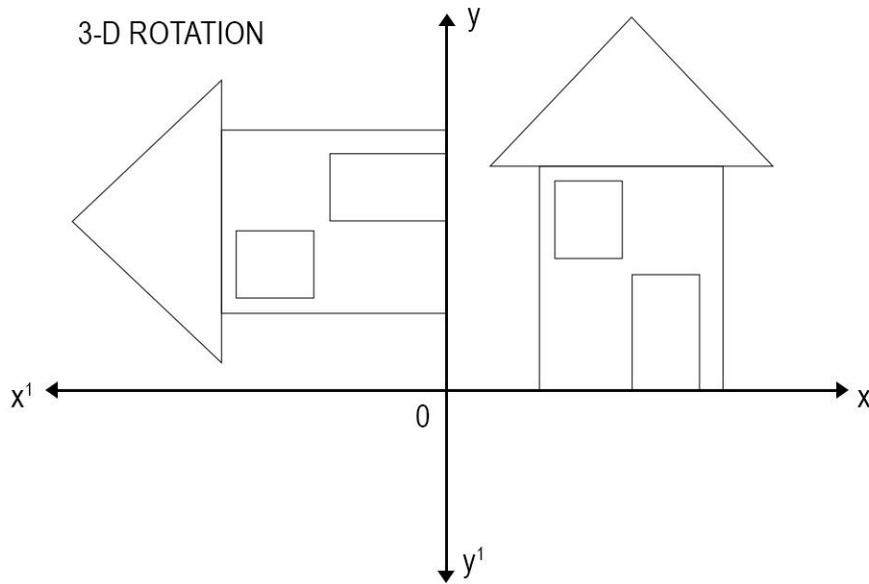


Fig 6.4.2 3D Rotation

6.4.3 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
- The following matrix shows scaling, $P'=P.S$

$$\text{Scaling x, y and z direction } P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} . S \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \end{bmatrix}$$

$$\text{Uniform Scaling x, y and z direction } P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} . S \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Non-Uniform Scaling x, y and z direction } P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} . S \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

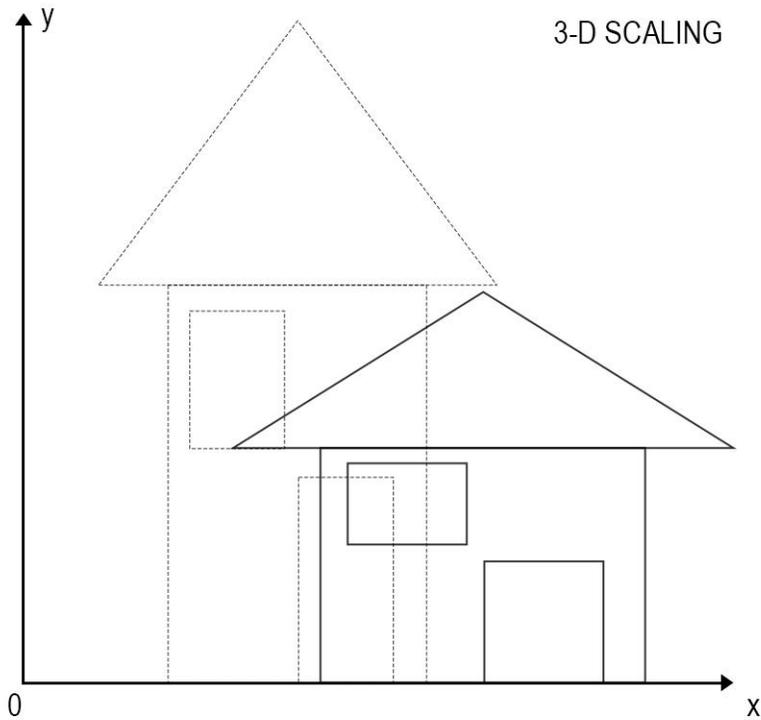


Fig 6.4.3 3D Scaling

6.4.4 Shearing:

- A shear is something that pushes the object sideways and make a thing little bit tilted.
- To achieve shearing an object should tilted(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.
- The following matrix shows shearing, $P'=P+Sh$

Shearing x, y and z direction
$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} 1 & a & b & 0 \\ c & 1 & d & 0 \\ e & f & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shearing in xy direction
$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shearing in yz direction
$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} 1 & a & b & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shearing in xz direction

$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

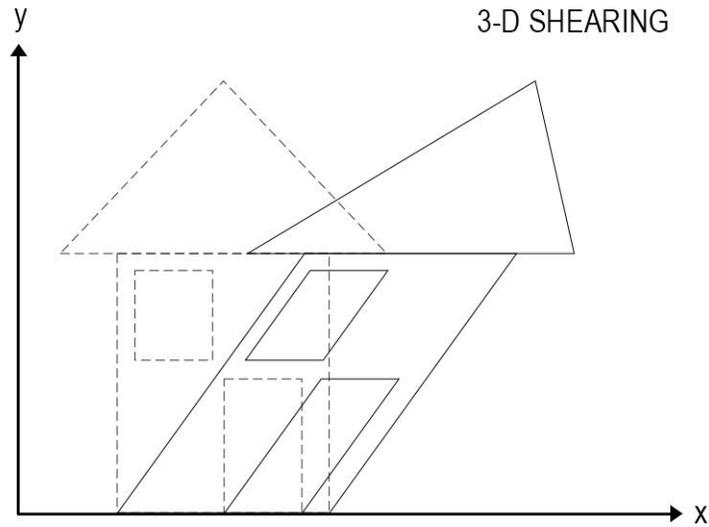


Fig 6.4.4 3D Shearing

6.4.5 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
- The following matrix shows reflection, $P'=P+RE_{xy}$

Reflection xy direction

$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection xz direction

$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection yz direction

$$P' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection about origin $p' \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot S \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

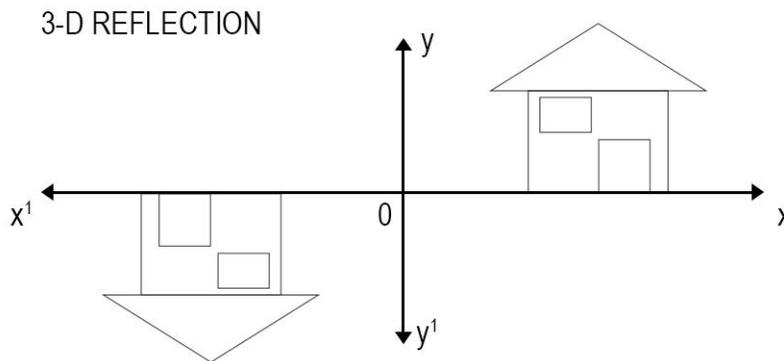


Fig 6.4.5 3D Reflection

6.5 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.

6.5.1 Method:

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

Example: Apply combined transformation on $\triangle ABC$, with $A = (2, 5)$, $B = (2, 1)$, $C = (5, 1)$ apply the following sequence of transformation on $\triangle ABC$

- Rotation by angle $\theta = 90^\circ$
- Reflection through x axis

Solution: The rotation matrix $\theta = 90^\circ$ ref fig 5.4.4

$$[R'] = \begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

The reflection through x axis matrix

$$[RE_x'] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The combined transformation matrix is

$$[R'] \cdot [RE_x'] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

Apply combined transformation on original points

$$[P'] = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \cdot ([R'] \cdot [RE_x']) = \begin{bmatrix} 2 & 5 \\ 2 & 1 \\ 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$[P'] = \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \begin{bmatrix} -2 & -5 \\ -2 & -1 \\ -5 & -1 \end{bmatrix}$$

6.6 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 $h \neq 0$

Translation: $[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}$

Rotation: $[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}$

Rotation: $[R_y] = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$

Rotation: $[R_z] = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Scaling: $[S] = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \end{bmatrix}$

$$\text{Shearing: } [Sh] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Reflection: } [RE.x] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6.7 ROTATION ABOUT AN ARBITRARY AXIS IN SPACE

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

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a_x & -b_y & -c_z & 1 \end{bmatrix}$$

2. Perform specified rotation about an arbitrary axis x and y. Rotation in anticlockwise angle implies positive direction therefore value of θ is positive

Rotation about an arbitrary x axis (anticlockwise rotation)

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}$$

Rotation about an arbitrary y axis (clockwise rotation)

$$[R_y] = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

3. Rotation about an angle θ in z axis $[R_z] = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
4. Apply Inverse rotation on step 2
5. Inverse Translation of step 1 to make an arbitrary axis in a particular direction except towards origin.

6.8 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 is required.

Step 1: Translation: translate the point P which 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 in step 2. $[R]^{-1}$

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

6.9 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.

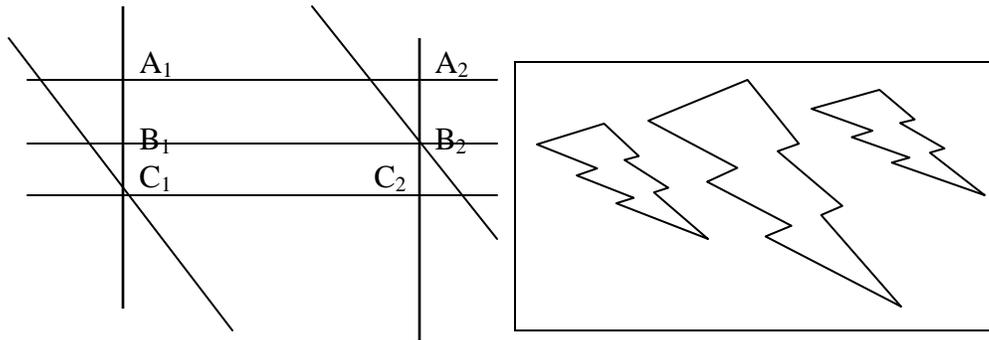


Fig 6.8.1 affine transformation and geometry

6.10 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. Perspective transformation represented in homogeneous coordinate system is as

follows in 4 X 4 matrix $[P_T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ where r is any non zero

value

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.

SINGLE POINT PERSPECTIVE

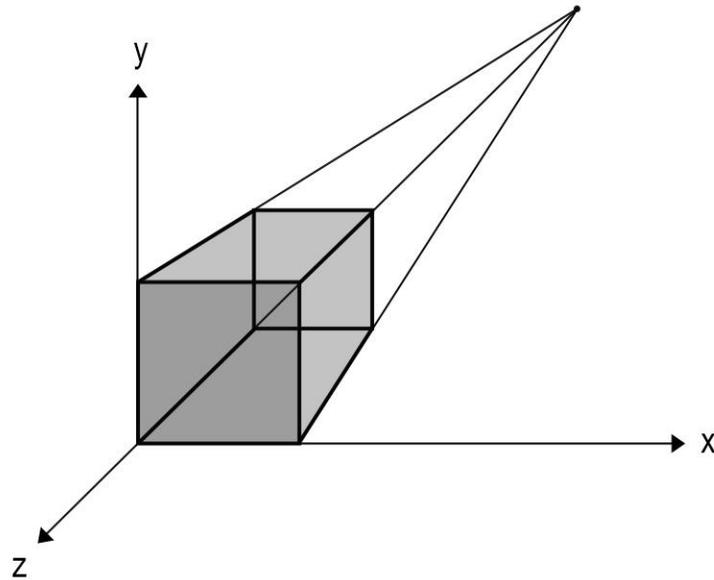


Fig 6.10.1 Single point perspective transformation

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.

TWO POINT PERSPECTIVE

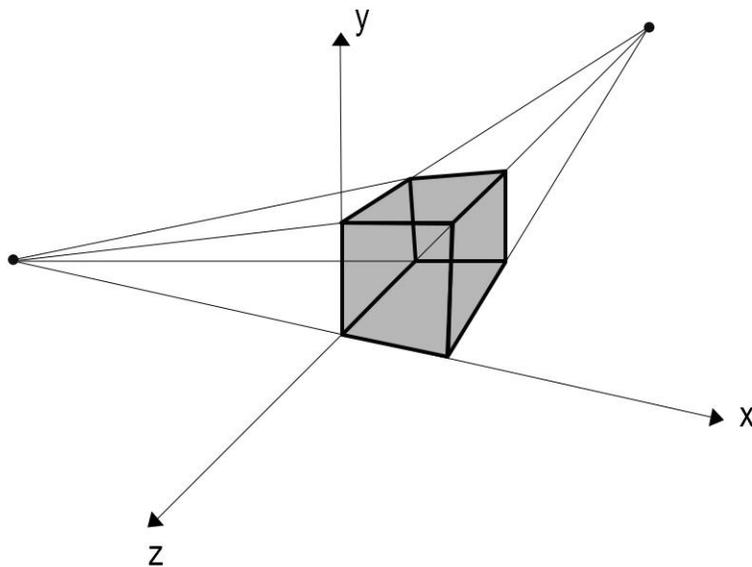


Fig 6.10.2 Two point perspective transformation

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.

THREE POINT PERSPECTIVE

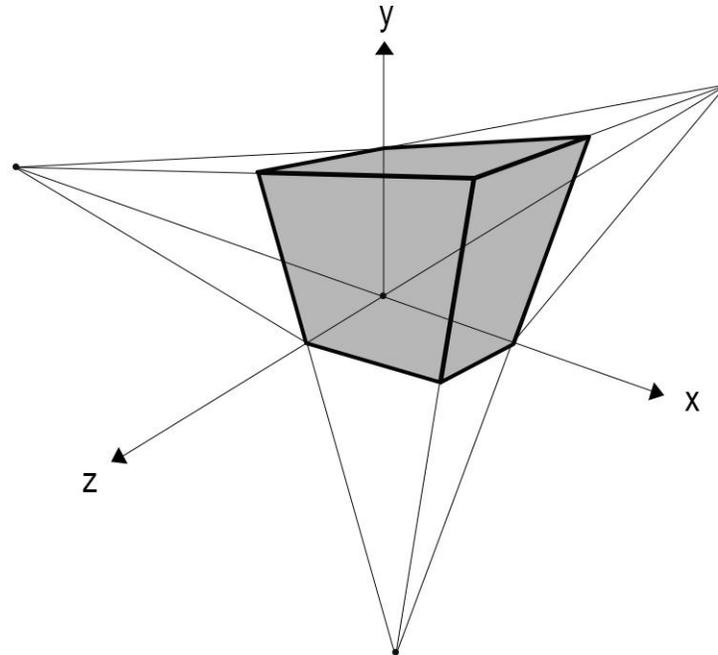


Fig 6.10.3 Three point perspective transformation

6.11 TECHNIQUES FOR GENERATING PERSPECTIVE VIEWS

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

1. Translation of object
2. Rotation

6.12 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.

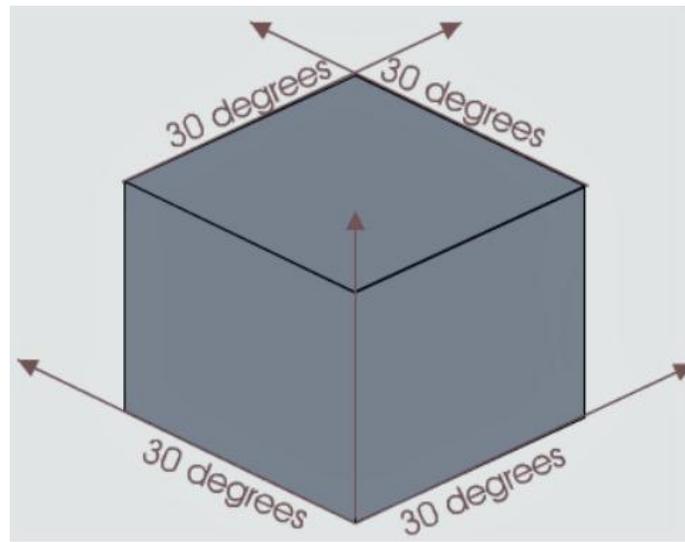
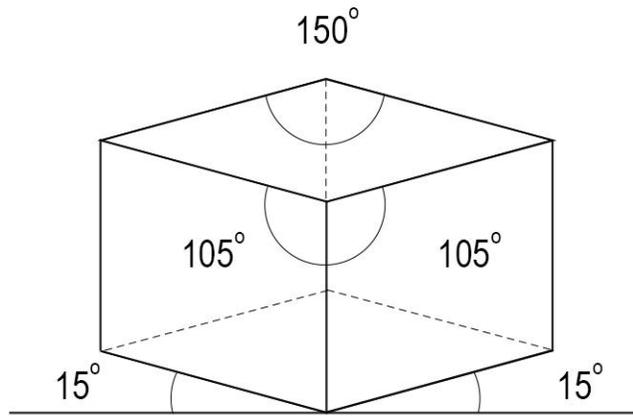


Fig 6.12.1 Isometric Projection

2. Dimetric projection: In isometric projection all three axis appears to be equally distanced but in Diometric 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.



DIMETRIC PROJECTION

Fig 6.12.2 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.

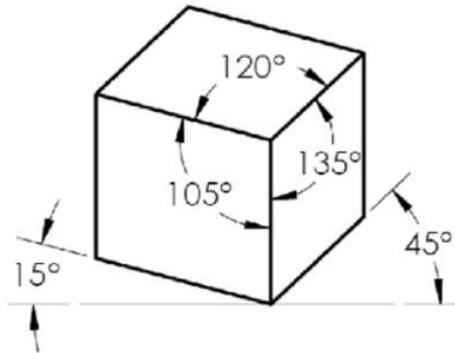


Fig 6.12.3 Trimetric Projection

6.13 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.

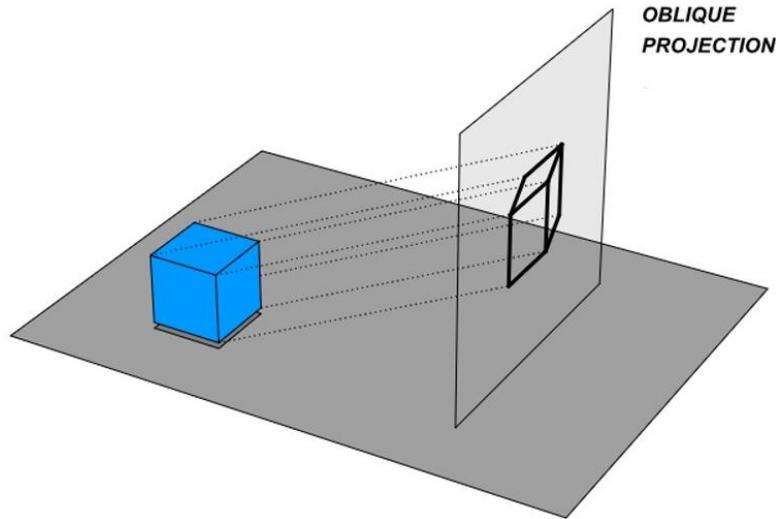


Fig 6.13.1 Oblique projection

1. Cavalier projection: Whenever the objects' edge parallel to the coordinate 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.

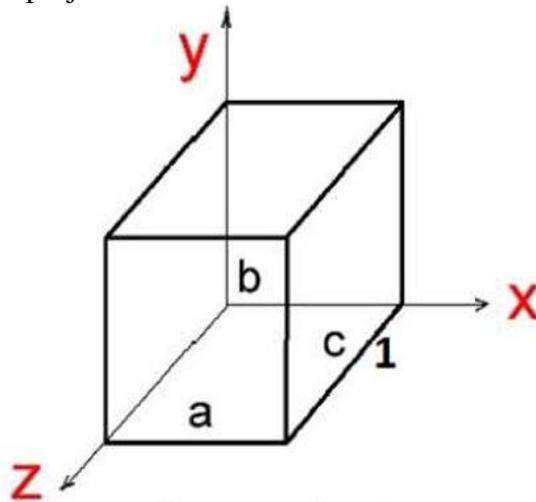


Fig 6.13.2 Cavalier Projection

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° .

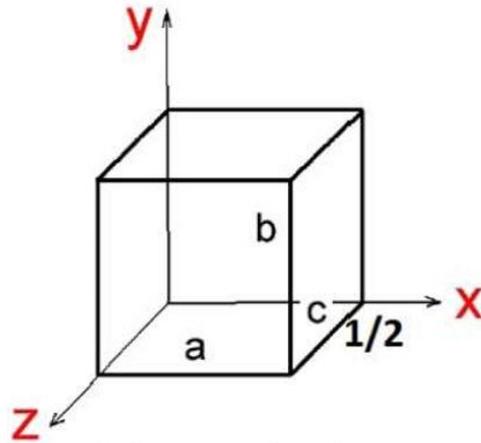


Fig 6.13.3 Cabinet Projection

6.14 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.

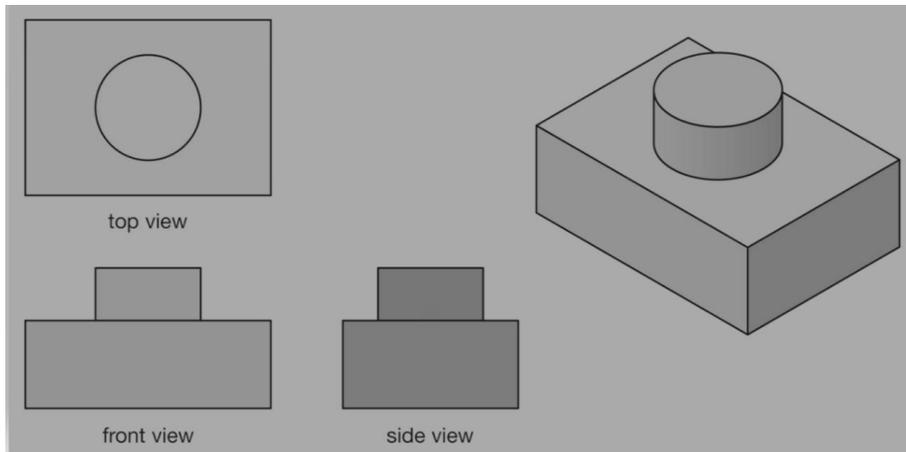


Fig 6.14.1 Orthographic Projection

6.15 VANISHING POINTS

A vanishing point is abstract point on the image plane where 2D projections 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 one point 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 some point 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 view plane at single point.

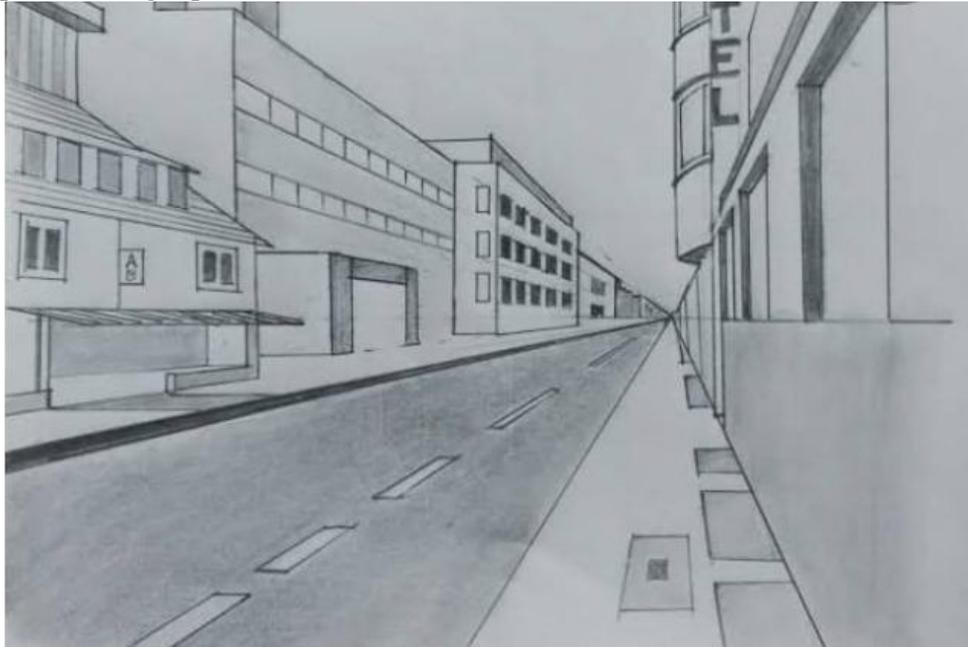


Fig 6.15.1 one point perspective vanishing point

2. Two point perspective:

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



Fig 6.15.2 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.

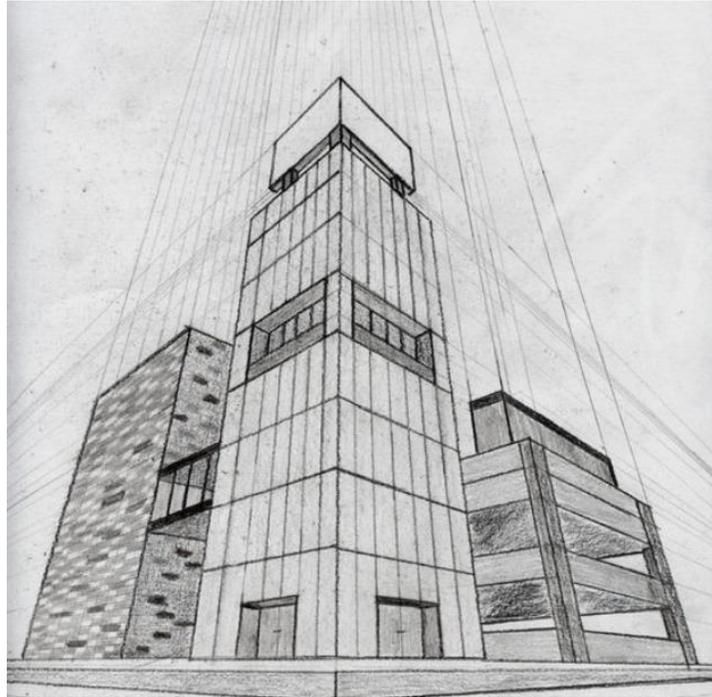


Fig 6.15.3 Three point perspective vanishing point

6.16 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 along parallel lines that are perpendicular to the projection plane.

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

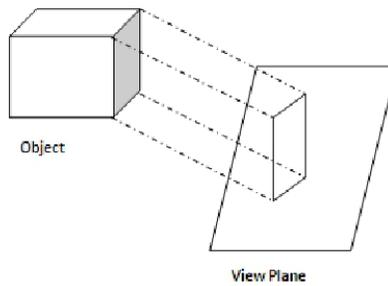


Fig 6.16.1 Parallel projection

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.

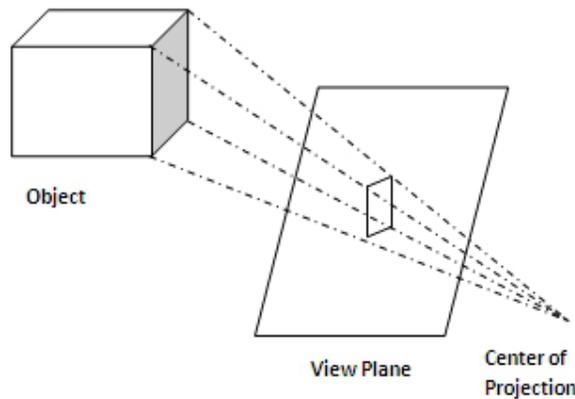


Fig 6.16.2 Perspective projection

View Volume:

A view volume is rectangular area in the view plane which controls how much of the scene is viewed. It 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.

6.17 SUMMARY

The process of changing size, shape, position and orientation of object is called transformation. The basic transformation is translation, rotation, reflection, scaling, shearing. All the transformation which are done in 2D that are same in 3D with the axis x, y, z. 3D matrix are represented by 4X 4 matrix with or without homogenous coordinate system. Rotation about an arbitrary axis can be done through translation and rotation. It can be done in both directions clockwise and anticlockwise. Reflection through an arbitrary plane is done with translation, rotation, reflection, inverse rotation and inverse translation. Study of parallel lines is called as affine and perspective geometry. Affine transformation can get through combination of two or more transformation. Perspective transformation deals with conventions of 3D world into 2D image. There are three types of perspective transformations single point, two point and three point perspective transformation. There are three ways to generating perspective views one point, two point and three point perspectives. It is also done in vanishing point in different perspective projection.

6.18 UNIT END EXERCISE

1. What is 3D transformation? Explain its types.
2. Explain conventions 3D transformation in detail.
3. Explain homogeneous coordinate system in 3D detail.
4. What are multiple 3D transformations? Explain in detail.
5. How many perspective vanishing points are there? How it works?
6. Write a short note on rotation about an arbitrary axis.
7. Explain in brief scaling in 3D.
8. Write a short note on cavalier and cabinet projection.
9. Explain reflection through an arbitrary plane.
10. Explain the concept of oblique projection in detail.

6.19 REFERENCES

- Computer Graphics C version 2nd Edition by Donald D. Hearn and M. Pauline Baker
- Computer Graphics A programming approach 2nd Edition by Steven Harrington McGraw Hill
- Fundamental of Computer Graphics 3rd Edition by Peter Shirley an Steven Marschner

UNIT III

7

VIEWING IN 3D

Viewing in 3D:- Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, The Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid.

Unit Structure

- 7.0 Objectives
- 7.1 Stages in 3D viewing
- 7.2 Canonical View Volume
- 7.3 Specifying an Arbitrary 3D View
- 7.4 Examples of 3D Viewing
- 7.5 The Mathematics of Planar Geometric Projections
- 7.6 Combined transformation matrices for projections and viewing
- 7.7 Coordinate Systems and matrices
- 7.8 Camera model and viewing pyramid
- 7.9 Summary
- 7.10 Unit End Exercises
- 7.11 List of References

7.0 OBJECTIVES

After going through this chapter, you will be able to:

- define 3D Viewing
- state the Stages in 3D viewing
- specify 3D Arbitrary view
- illustrate the 3D Viewing
- describe the mathematics of planar geometric projections
- derive combined transformation matrices for projections and viewing
- classify different types of Coordinate Systems and matrices
- explain Camera model and viewing pyramid

7.1 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

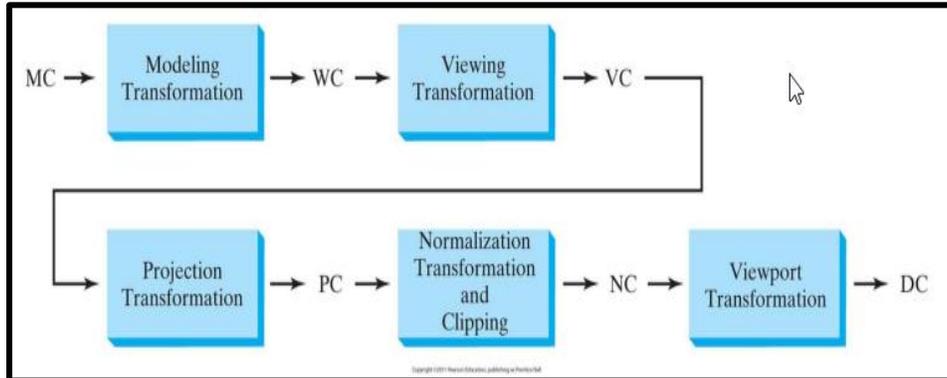


Fig: 7.1 Stages in 3D Viewing

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 output device.
- It transforms Normalized Coordinates into viewport is 2D Device coordinates (DC) or display.

7.2 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:**

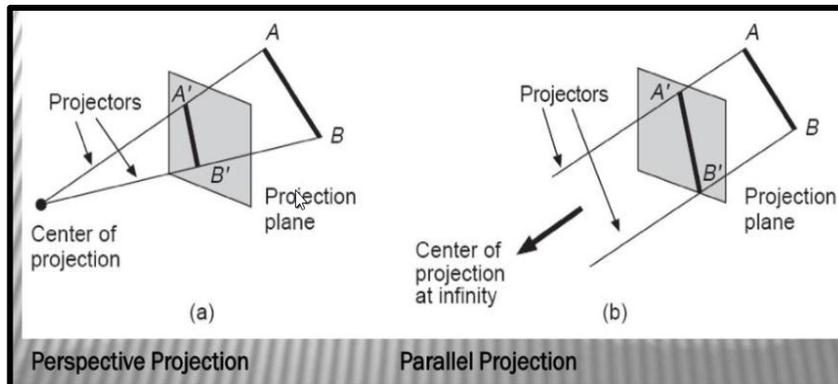


Figure 7.2: Perspective and Parallel (Orthographic) projection for CVV demonstrations

Both types of projections, transform a given viewing volume to the canonical viewing volume (CVV):

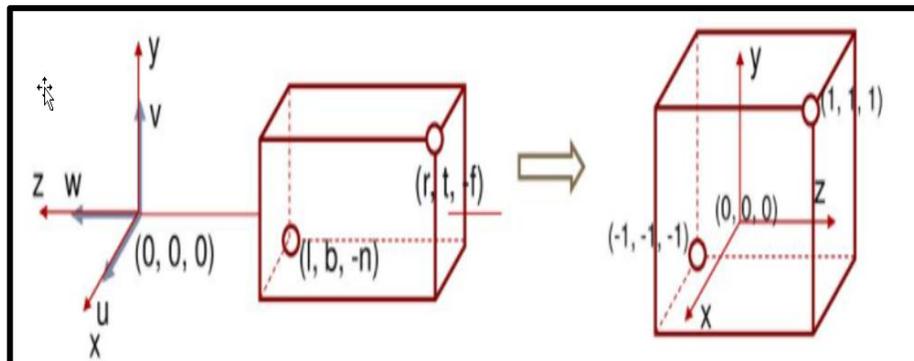


Fig: 7.3

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.

This is accomplished by the following matrix,

$$M_{orth} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7.3 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 common way to specify an arbitrary view is to specify the following:

- ❖ A View Reference Point (VRP) which is to point on the plane.
- ❖ A View Plane Normal (VPN) which is the normal vector to the plane.
- ❖ A View Up Vector (VUP) which is a vector from which we determine 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, VPN and VUP

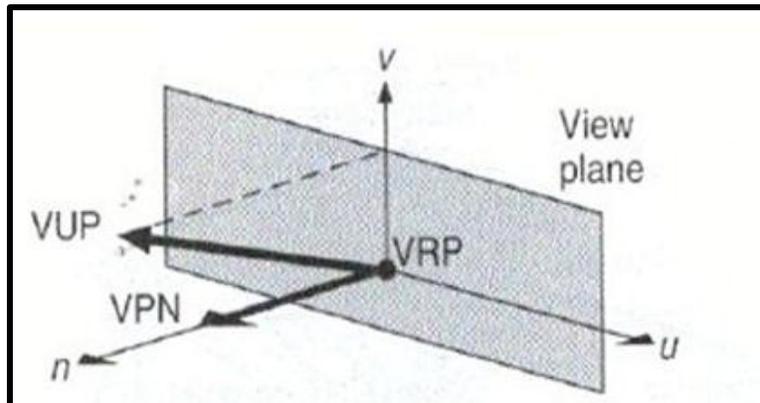


Figure 7.4: The view plane is defined by VPN and VRP, the v axis is defined by the projection of VUP along VPN onto the view plane. The u axis forms the right-handed VRC system with VPN and v.

7.4 EXAMPLES OF 3D VIEWING

- In 3D viewing examples, we are going to set value for VRP, VPN, VUP, PRP, VRC and F and B of VRC based on the type of projection along with x, y and z principal axes.

Where,

VRP → View reference point

VPN → View plane normal

VUP → View up vector

PRP → Projection Reference point

VRC → Viewing reference co-ordinate

F and B of VRC → front and back of VRC

- 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 defaults are shown in below figure **7.5(a)**. The view volume corresponding to these defaults is shown in figure **7.5(b)**. If the type of projection is perspective rather than parallel, then the view volume is the pyramid as shown in **7.5(c)**.

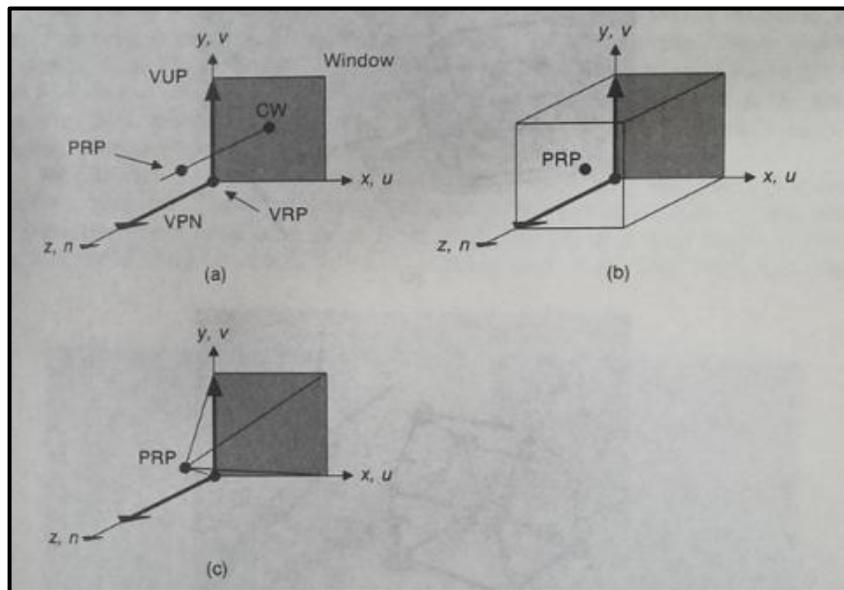


Fig: 7.5 (a) The default viewing specification: VRC is at the origin, VUP is the y axis, and VPN is the z -axis. This makes the VRC system of u, v and n coincide with x, y, z world-coordinate system. The window extends from 0 to 1 along u and v , and PRP is at (0.5, 0.5, and 1.0). (b) Default parallel projection view volume. (c) View volume if default projection were perspective.

- 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	

- Readers wanting to review how all these parameters interconnect are encouraged to construct a house, the world coordinate system, and the VRC system with Tinker Toys, as pictured in **Fig. 7.6**. The idea is to position the VRC system in world coordinates as in the viewing example and to imagine projectors from points on the house intersecting the view plane. In our experience, this is a valuable way to know 3D viewing concepts.

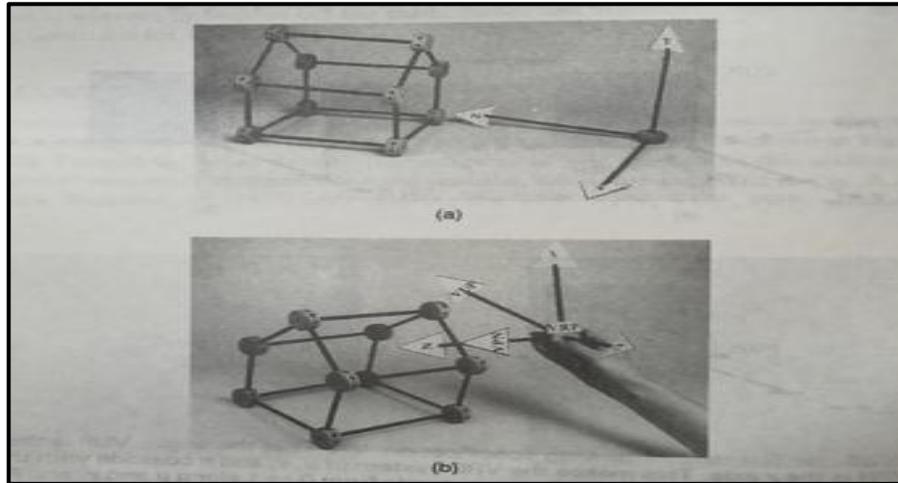


Fig: 7.6 Stick models useful for understanding 3D Viewing. (a) House and World-coordinate system. (b) House and VRC system

7.4.1 Perspective Projections

- To obtain the front one-point perspective view of the house shown in **Fig. 7.7** we position the center of projection (which can be thought of as the position of the viewer) at $x = 8$, $y = 6$, and $z = 84$. 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 z value 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 **Fig: 7.7** 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 **Fig: 7.8** 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.
- We place the view plane on the front face of the house by placing the VRP anywhere in the: = 54 plane; (0, 0, 54), the lower-left front corner of the house, is fine. For the center of projection to be the same as in **Fig: 7.7**, the PRP, which is in the VRC system, needs to be at (8, 6, 30). **Fig: 7.9** shows this new arrangement of the VRC, VRP and PRP,

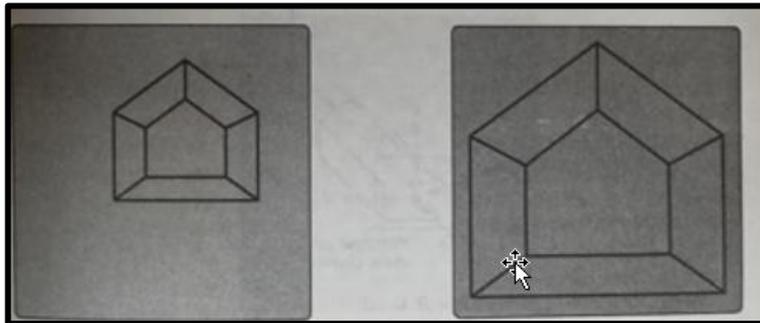


Fig: 7.7 One-point perspective projection of the house & **Fig: 7.8** Centered perspective projection of a house

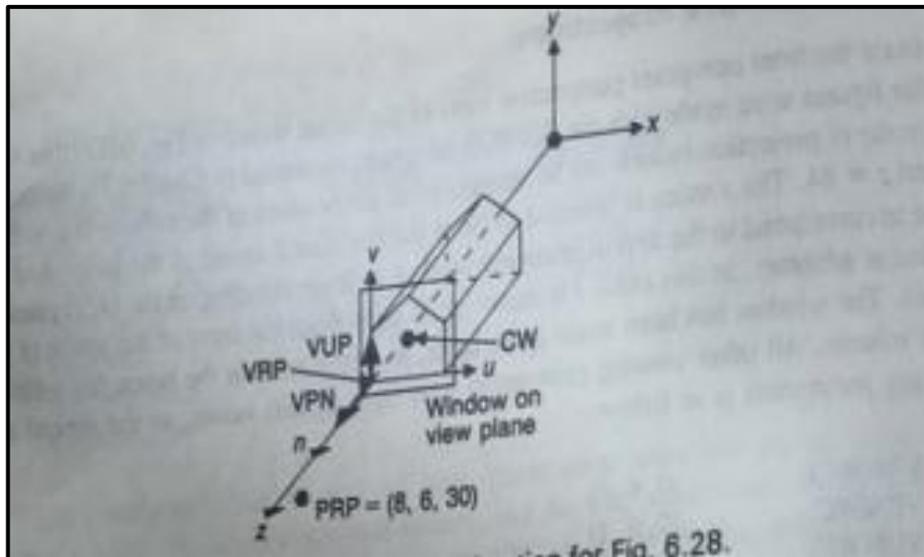


Fig: 7.9 The viewing situation for the **figure 7.8**

which corresponds to the following set of viewing parameters:

VRP(WC)	(0, 0, 54)
VPN(WC)	(0, 0, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(8, 6, 30)
window(VRC)	(-1, 17, -1, 17)
projection type	perspective

- This same result can be obtained in many other ways. For instance, with the VRP at (8, 6, 54), as in **Fig. 7.10**, the center of projection, given by the PRP, becomes (0, 0, 30). The window must also be changed, because its definition is based on the VRC system, the origin of which is the VRP. The appropriate window extends from -9 to 9 in u and from -7 to 11 in v . With respect to the house, this is the same window as that used in the previous example, but it is now specified in a different VRC system. Because the view-up direction is the y axis, the u axis and x axis are parallel, as are the v and y axes. In summary, the following viewing parameters, shown in **Fig. 7.10**, also produce **Fig. 7.8**:

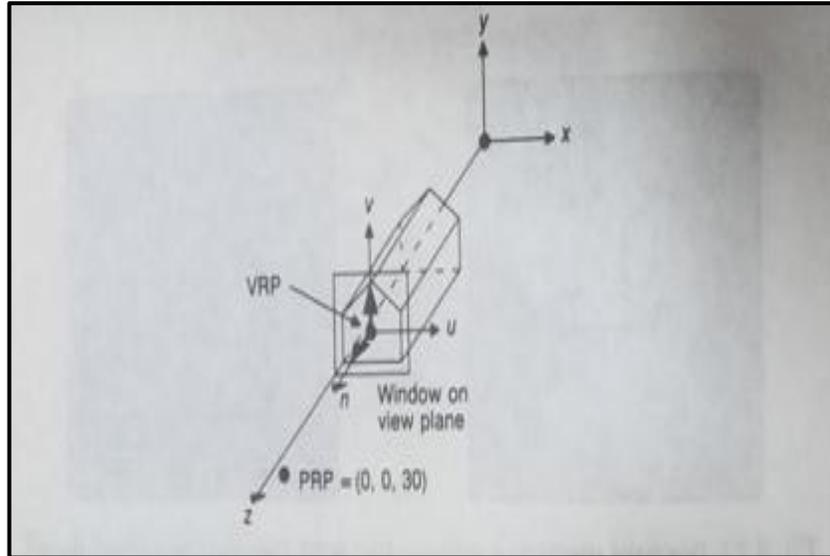


Fig: 7.10 An alternative viewing situation for fig. 7.8.

VRP(WC)	(8, 6, 54)
VPN(WC)	(0, 0, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(0, 0, 30)
window(VRC)	(-9, 9, -7, 11)
projection type	perspective

- Next, for obtaining the two-point perspective projection as shown in **Fig. 7.11**. The center of projection is analogous to the position of a camera that takes snapshots of world-coordinate objects. With this analogy in mind, the center of projection in **Fig. 7.11** seems to be somewhat above and to the right of the house, as viewed from the positive z axis. The exact center of projection is (36, 25, 74).

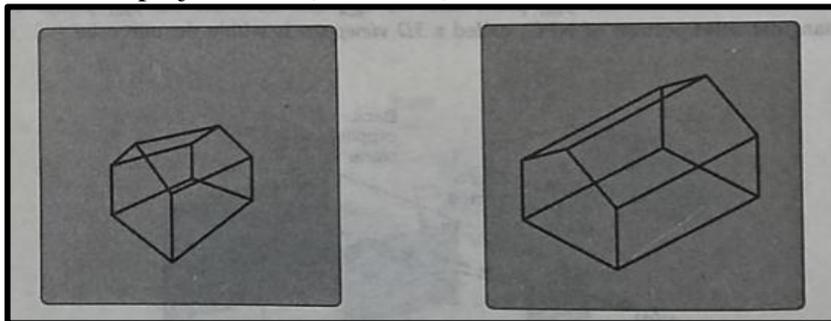


Fig: 7.11 Two-point perspective projection of a house and Fig: 7.12 Isometric projection of a house

- Now, if the corner of the house at (16, 0, 54) is chosen as the VRP, then this center of projection is at (20, 25, 20) relative to it. With the view plane coincident with the front of the house (the $z = 54$ plane), a window ranging from -20 to 20 in u and from -5 to 35 in v is certainly large

enough to contain the projection. Hence, we can specify the view of **Fig. 7.13** with the viewing parameters:

VRP(WC)	(16, 0, 54)
VPN(WC)	(0, 0, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(20, 25, 20)
window(VRC)	(-20, 20, -5, 35)
projection type	perspective

- This view is similar to, but clearly is not the same as, that in **Fig. 7.11** as is a two-point perspective projection, whereas **Fig. 7.13** is a one-point perspective. It is apparent that simply moving the center of projection is not sufficient to produce **Fig. 7.11**. In fact, we need to reorient the view plane such that it cuts both the x and z axes, by setting VPN to (1, 0, 1).

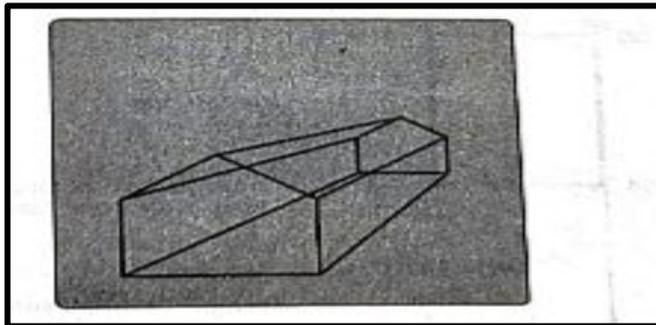


Fig: 7.13 Perspective projection of a house from (36, 25, 74) with VPN parallel to the z axis.

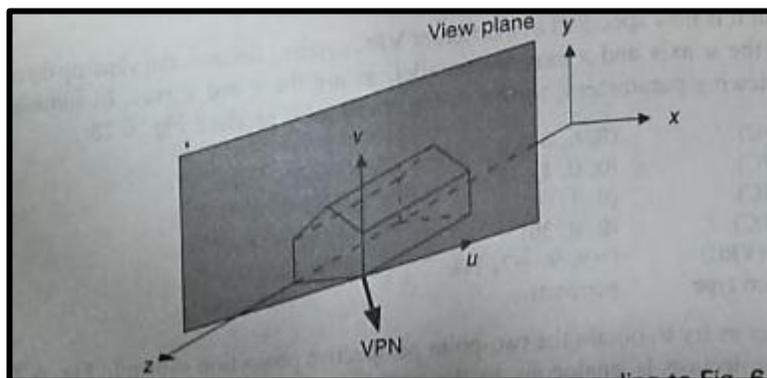


Fig: 7.14 The view plane and VRC system corresponding to **Fig. 7.11**

Thus, the viewing parameters for **Fig. 7.11** are as follows:

VRP(WC)	(16, 0, 54)
VPN(WC)	(1, 0, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(0, 25, $20\sqrt{2}$)
window(VRC)	(-20, 20, -5, 35)
projection type	perspective

- **Figure 7.11** shows the view plane established with this VPN. There are two ways to choose a window that completely surrounds the projection, as does the window in **Fig. 7.11**. One can estimate the size of the projection of the house onto the view plane as shown in **Fig. 7.15**, to calculate the intersections of projectors.

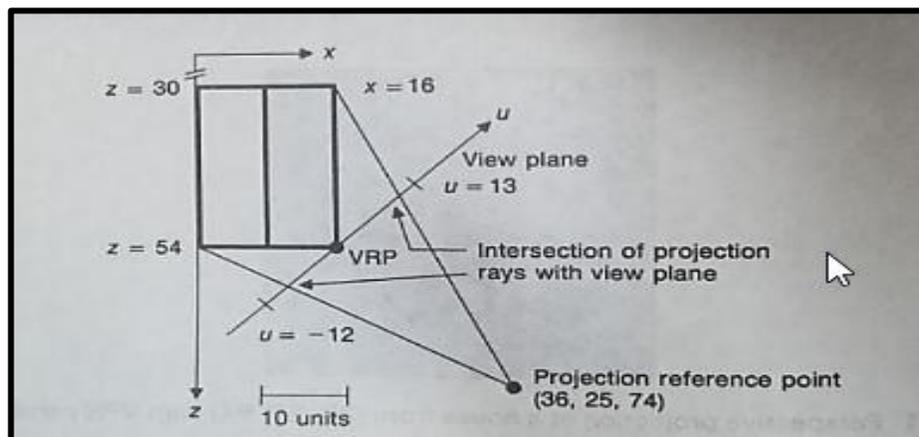


Fig: 7.15 Top (plan) view of a house for determining an appropriate window size

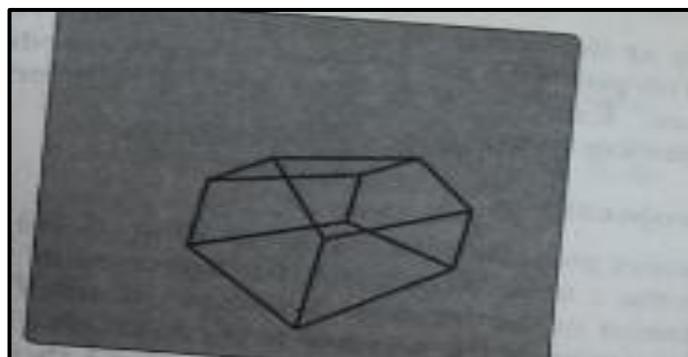


Fig: 7.16 Projection of house produced by rotating VUP.

with the view plane. A better alternative, however, is to allow the window bounds to be variables in a program that are determined interactively via a valuator or locator device. **Figure 7.16** is obtained from the same projection as is **Fig. 7.11** but the window has a different orientation.

Fig 7.16 has exactly the same viewing parameters as does **Fig. 7.11**, except that VUP has been rotated away from the y axis by about 10° .

- Another way to specify viewing parameters for perspective projections is suggested in **Fig. 7.17**. This figure is showed after the way a photographer might reason about placing a camera.
- **Six parameters are needed:** the center of projection, which is the camera position; the center of attention, which is a point at which the camera is meant (the VPN is the vector from the center of attention to the center of projection); VUP, the up vector, D, the distance from the center of projection to the projection plane; W, the width of the window on the projection plane, and H, the height of the window on the projection plane.
- The center of attention need not be on the view plane. In this model, the VPN is always pointed directly at the center of projection, and the view volume is symmetrical about its center line.

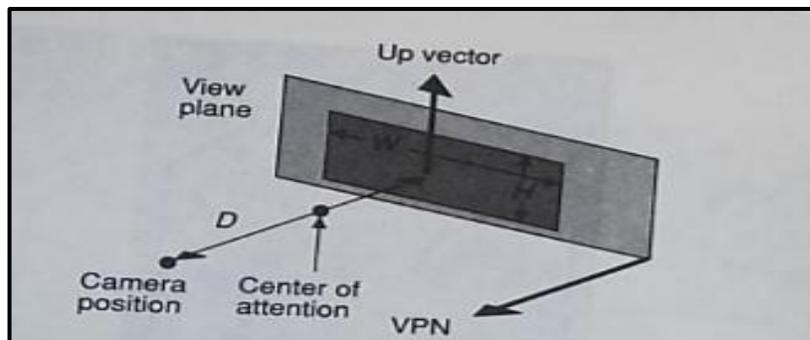


Fig. 7.17 Another way to specify a view, with a camera position (the center of projection), center of attention, up vector, distance from the center of projection to the projection plane, and the height and width of the window on the projection plane. VPN is parallel to the direction from the center of attention to the camera position.

7.4.2 Parallel Projections:

- We create a front parallel projection of the house (**Fig. 7.18**) 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 $(8, 8, 0)$.
- A PRP of $(8, 8, 100)$ offers a direction of projection parallel to the z axis. **Fig. 7.19** shows the viewing situation that creates **Fig. 7.18**.

- The viewing parameters are as follows:

VRP(WC)	(0, 0, 0)
VPN(WC)	(0, 0, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(8, 8, 100)
window(VRC)	(-1, 17, -1, 17)
projection type	parallel

- To create the side view (**Fig. 7.19**), we require the viewing situation of **Fig. 7.20**, with the (y, z) plane (or any plane parallel to it) as the view plane. This situation corresponds to the following viewing parameters:

VRP(WC)	(0, 0, 54)	x axis
VPN(WC)	(1, 0, 0)	y axis
VUP(WC)	(0, 1, 0)	
PRP(VRC)	(12, 8, 16)	
window(VRC)	(-1, 25, -5, 21)	
projection type	parallel	

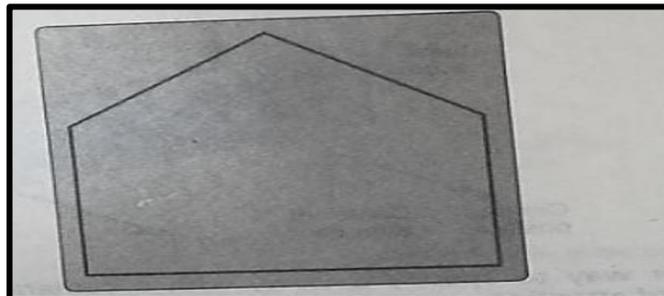


Fig. 7.18 Front parallel projection of the house.

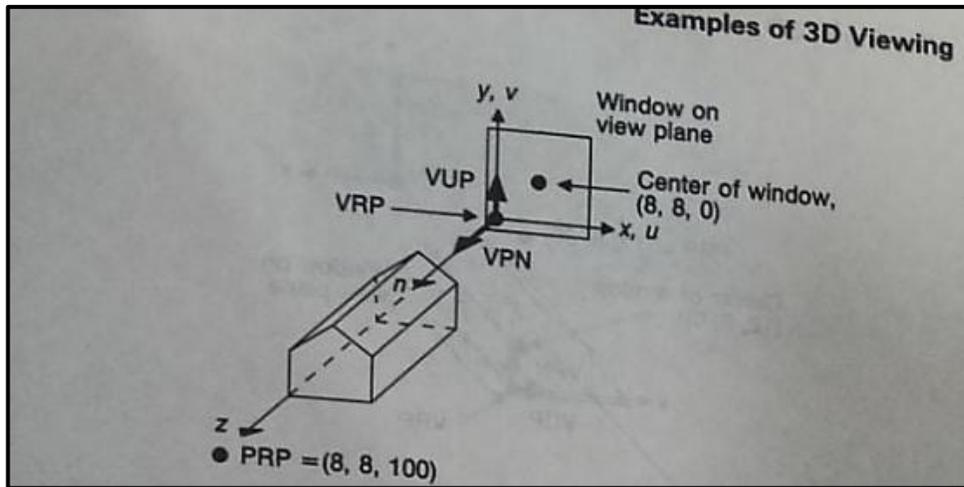


Fig. 7.19 Viewing situation that creates Fig. 7.18, a front view of the house. The PRP could be any point with $x = 8$ and $y = 8$.

- The center of the window is at $(12, 8, 0)$ in VRC; hence, the PRP has these same u and v coordinates.
- We create a top view of the house by using the (x, z) plane as the view plane and VPN as the y axis. The default view-up direction of $+y$ must be changed; we use the negative x axis. With VRP again stated as a corner of the house, we have the viewing situation in Fig. 7.22, defined by the following viewing parameters:

VRP(WC)	(16, 0, 54)	
VPN(WC)	(0, 1, 0)	y axis
VUP(WC)	(-1, 0, 0)	negative x axis
PRP(VRC)	(12, 8, 30)	
window(VRC)	(-1, 25, -5, 21)	
projection type	parallel	

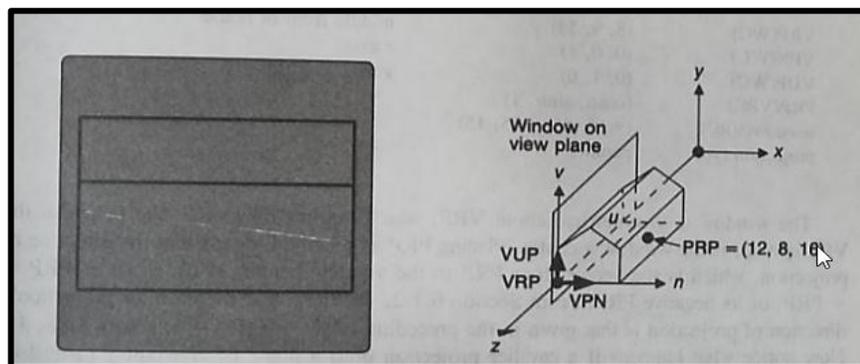


Fig: 7.20 Parallel projection from the side of the house & Fig: 7.21 The viewing situation for Fig: 7.20

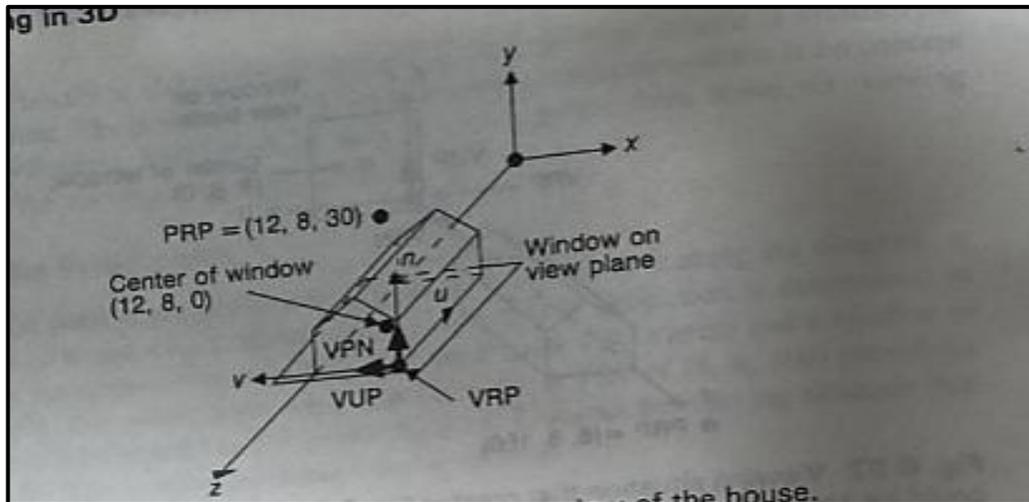


Fig: 7.22 The viewing situation for a top view of the house.

- **Figure 7.12** is an isometric (parallel orthographic) projection in the direction $(-1, -1, -1)$, one of the eight possible directions for an isometric. The following viewing parameters create such an isometric projection:

VRP(WC)	(8, 8, 42)
VPN(WC)	(1, 1, 1)
VUP(WC)	(0, 1, 0)
PRP(VRC)	(0, 0, 10)
window(VRC)	(-20, 20, -20, 20)
projection type	parallel

- A cavalier projection with **angle α** is specified with

VRP(WC)	(8, 8, 54)	middle front of house
VPN(WC)	(0, 0, 1)	z axis
VUP(WC)	(0, 1, 0)	y axis
PRP(VRC)	($\cos\alpha, \sin\alpha, 1$)	
window(VRC)	(-15, 15, -15, 15)	
projection type	parallel	

- The window is symmetrical about VRP, which implies that VRP, the origin of the VRC system, is the window's center. Placing PRP as specified means that the direction of projection, which is the vector from PRP to the window center, is $(0, 0, 0) - \text{PRP} = -\text{PRP}$, or its negative PRP.

7.4.3 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 .

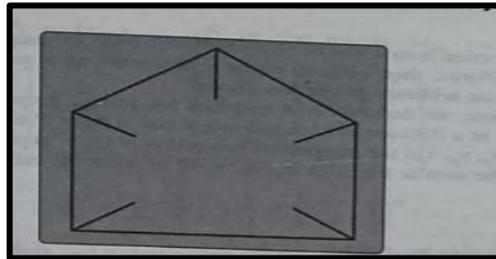


Fig: 7.23 Perspective projection of the house with back clipping plane at $z = 31$.

- A front perspective view of the house with the rear wall clipped away (**Fig: 7.23**) results from the following viewing specification, in which F and B have been added.
- If a distance is given, then clipping against the corresponding plane is assumed; otherwise, it is not.
- The **viewing specification is as follows:**

VRP(WC)	(0, 0, 54)	lower-left front of house
VPN(WC)	(0, 0, 1)	z axis
VUP(WC)	(0, 1, 0)	y axis
PRP(VRC)	(8, 6, 30)	
window(VRC)	(-1, 17, -1, 17)	
projection type	perspective	
F (VRC)	+1	one unit in front of house, at $z = 54 + 1 = 55$
B (VRC)	-23	one unit from back of house, at $z = 54 - 23 = 31$

- 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.

7.5 THE MATHEMATICS OF PLANAR GEOMETRIC PROJECTIONS

- Planar projections are the subset of 3D graphical projections constructed by linearly mapping points in three-dimensional projection plane. 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 acting on 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 4×4 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 4×4 matrices for several projections, beginning with a projection plane at a distance d from the origin and a point \mathbf{P} to be projected onto it.
- To calculate $\mathbf{P}_p = (x_p, y_p, z_p)$, the perspective projection of (x, y, z) onto the projection plane at $z = d$, we use the similar triangles in **Fig: 7.24** to write the ratios

$$\frac{x_p}{d} = \frac{x}{z}; \quad \frac{y_p}{d} = \frac{y}{z} \text{----- (7.1)}$$

Multiplying each side by d yields

$$x_p = \frac{d \cdot x}{z} = \frac{x}{z/d} \quad y_p = \frac{d \cdot y}{z} = \frac{y}{z/d} \text{..... (7.2)}$$

- The distance d is just a scale factor applied to x_p and y_p . The division by z causes the perspective projection of more distant objects to be smaller than that of closer objects. All values of z are permissible except $z = 0$. Points can be behind the center of

projection on the negative z axis or between the center of and the projection plane. The transformation of **Eq. (7.2)** can be expressed as a 4 x 4 matrix:

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \dots\dots\dots (7.3)$$

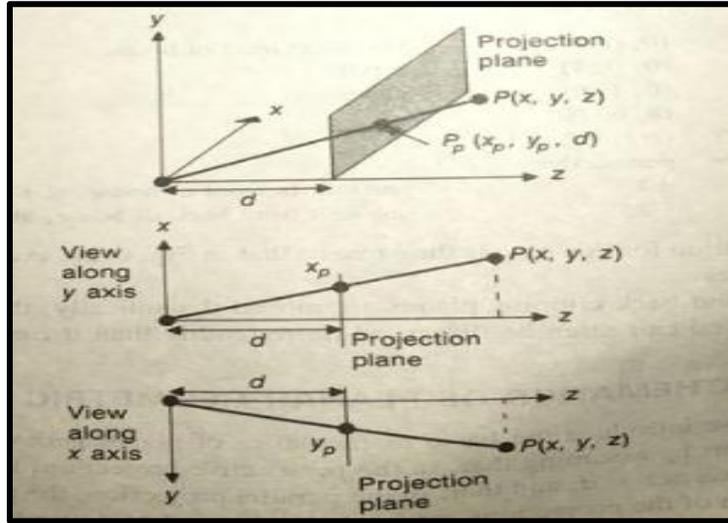


Fig: 7.24 Perspective Projection

- Multiplying the point $P = [x \ y \ z \ 1]^T$ by the matrix M_{per} yields the general homogeneous point $[X \ Y \ Z \ W]^T$:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_{per} \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \dots\dots\dots (7.4)$$

Or

$$[X \ Y \ Z \ W]^T = [x \ y \ z \ \frac{z}{d}]^T \dots\dots\dots (7.5)$$

- Now, dividing by W (which is z/d) and dropping the fourth coordinate to come back to 3D we have

$$\left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W}\right) = (x_p, y_p, z_p) = \left(\frac{x}{z/d}, \frac{y}{z/d}, d\right); \dots\dots\dots (7.6)$$

- These equations are the correct results of **Eq. (7.1)**, plus the transformed z coordinate of d, which is the position of the projection plane along the z axis.
- An alternative design for the perspective projection places the projection plane at $z = 0$ and the center of projection at $z = -d$,

as in **fig: 7.25**. Similarity of the triangle now gives

$$\frac{x_p}{d} = \frac{x}{z+d}, \frac{y_p}{d} = \frac{y}{z+d} \text{ ----- (7.7)}$$

Multiplying by d, we get

$$x_p = \frac{d \cdot x}{z+d} = \frac{x}{z/d+1}, y_p = \frac{d \cdot y}{z+d} = \frac{y}{z/d+1} \text{ ----- (7.8)}$$

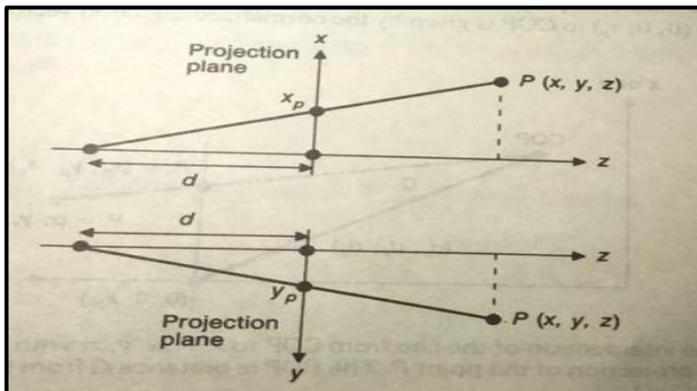


Fig.7.25: Alternative Perspective Projection

The matrix is

$$M'_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \text{ ----- (7.9)}$$

- This formulation allows d, the distance to the center of projection to tend to infinity.
- The orthographic projection onto a projection plane at $z = 0$ is straightforward. The direction of projection is the same as the projection-plane normal-the z axis in this case. Thus, point P projects as

$$x_p = x, y_p = y, z_p = 0 \text{ ----- (7.10)}$$

- This projection is expressed by the matrix

$$M_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ ----- (7.11)}$$

- Notice that as **d** in **Eq. (7.9)** tends to infinity, **Eq. (7.9)** becomes **Eq. (7.11)**.
- 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.
- In **Fig: 7.26**, the projection of the general point $P = (x, y, z)$ onto the projection plane is $P = (x_p, y_p, z_p)$. The projection plane is

perpendicular to the z axis at a distance z_p , from the origin, and the center of projection (COP) is a distance Q from the point $(0,0,z_p)$. The direction from $(0,0,z_p)$ to COP is given by the normalized direction vector (d_x, d_y, d_z) . P_p is on the line between COP and P, which can be stated parametrically as

$$COP + t(P - COP), \quad 0 \leq t \leq 1 \text{ ----- (7.12)}$$

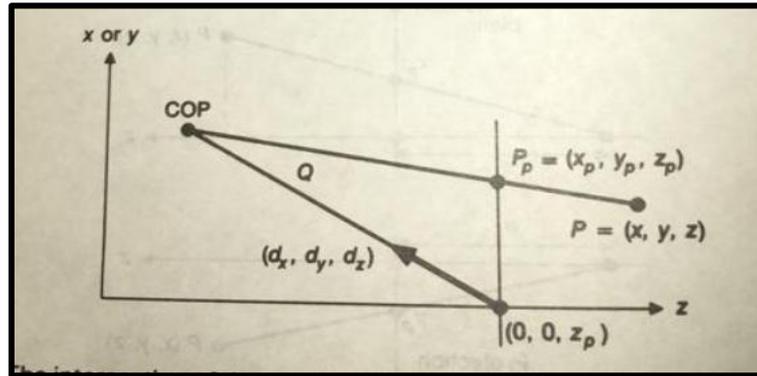


Fig: 7.26 - The intersection of the line from COP to $P=(x,y,z)$ with the projection plane at $z= z_p$ is the projection of the point P. The COP is distance Q from the point P . The COP is the distance Q from the point $(0, 0,z_p)$ in direction (d_x, d_y, d_z)

- Rewriting eq. (7.12) as separate equations for the arbitrary point $P' = (x', y', z')$ on the line with $COP = (0,0,z_p) + Q(d_x, d_y, d_z)$ yields

$$x' = Qd_x + (x - Qd_x)t, \quad \text{----- (7.13)}$$

$$y' = Qd_y + (y - Qd_y)t, \quad \text{----- (7.14)}$$

$$z' = (z_p + Qd_z) + (z - (z_p + Qd_z))t, \text{----- (7.15)}$$

- We find the projection P_p of the point P, at the intersection of the line between COP and P with the projection plane, by substituting $z' = z_p$ into **Eq. (7.15)** and solving for t ;

$$t = \frac{z_p - (z_p + Qd_z)}{z - (z_p + Qd_z)} \quad \text{----- (7.16)}$$

- Substituting this value of t into **Eq. (7.13)** and **Eq. (7.14)** to find $x' = x_p$ and $y' = y_p$ yields

$$x_p = \frac{x - z \frac{d_x}{d_z} + z_p \frac{d_x}{d_z}}{\frac{z_p - z}{Qd_z} + 1} \quad \text{----- (7.17)}$$

$$y_p = \frac{y - z \frac{d_y}{d_z} + z_p \frac{d_y}{d_z}}{\frac{z_p - z}{Qd_z} + 1} \quad \text{----- (7.18)}$$

- Multiplying the identity $z_p = z_p$ on the right-hand side by fraction whose numerator and denominator are both

$$\frac{z_p - z}{Qd_z} + 1 \text{ ----- (7.19)}$$

- maintains the identity and gives z_p the same denominator as x_p and y_p :

$$z_p = z_p \frac{\frac{z_p - z}{Qd_z} + 1}{\frac{z_p - z}{Qd_z} + 1} = \frac{-z \frac{z_p}{Qd_z} + \frac{z_p^2 + z_p Qd_z}{Qd_z}}{\frac{z_p - z}{Qd_z} + 1} \text{ ----- (7.20)}$$

- Now **Eqs. (7.17), (7.18) and (7.20)** can be rewritten as a 4x4 matrix $M_{general}$ arranged so that the last row of $M_{general}$ multiplied by $[x \ y \ z \ 1]^T$ produces their common denominator, which is homogeneous coordinate W and is hence the divisor of X, Y and Z:

$$M_{general} = \begin{bmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p \frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p \frac{d_y}{d_z} \\ 0 & 0 & -\frac{z_p}{Qd_z} & \frac{z_p^2}{Qd_z} + z_p \\ 0 & 0 & -\frac{1}{Qd_z} & \frac{z_p^2}{Qd_z} + 1 \end{bmatrix} \text{ ----- (7.21)}$$

- $M_{general}$ specialized to all three of previously derived projections matrixes M_{per} , M'_{per} and M_{ort} given the following values:

	z_p	Q	$[d_x \ d_y \ d_z]$
M_{ort}	0	∞	$[0 \ 0 \ -1]$
M_{per}	d	d	$[0 \ 0 \ -1]$
M'_{per}	0	d	$[0 \ 0 \ -1]$

- When $Q \neq \infty$, $M_{general}$ defines a one-point perspective projection. The vanishing point of a perspective projection is calculated by multiplying the point at infinity on the z axis, represented in homogenous coordinate as $[0 \ 0 \ 1 \ 0]^T$ by $M_{general}$.
- Taking this product and dividing by **W** gives

$$x = Qd_x, y = Qd_y, z = z_p$$
- Given a desired vanishing point (x,y) and known distance Q to the center of projection these equations uniquely define $[d_x \ d_y \ d_z]$, because $\sqrt{d_x^2 + d_y^2 + d_z^2} = 1$

- Similarly it is easy to show that for cavalier and cabinet projection onto the (x,y) plane with α the angle.

	z_p	Q	$[d_x \quad d_y \quad d_z]$
Cavalier	0	∞	$[\cos\alpha \quad \sin\alpha \quad -1]$
Cabinet	0	∞	$\left[\frac{\cos\alpha}{2} \quad \frac{\sin\alpha}{2} \quad -1\right]$

- In this section, we have seen how to formulate M_{per} , M'_{per} and M_{ort} , all of which are special cases of the more general $M_{general}$. In all these cases, however the projection plane is perpendicular to the z axis.

7.6 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 convert a point from world coordinates to Normalized Device Coordinates (NDC) while retaining a representation of the relative depth of the points.

$$MTOT = A_{wv} * M_p * N$$

Where,

MTOT: - Represents Combined transformation matrices for projection and viewing

A_{wv}: - Represents Metrics of real-world coordinates

M_p: - 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**.

7.7 COORDINATE SYSTEMS AND MATRICES

A Coordinate system is used to unambiguously represent point, it contains a reference point (the origin) and three linear independent vectors (the basis) in the below following figure:

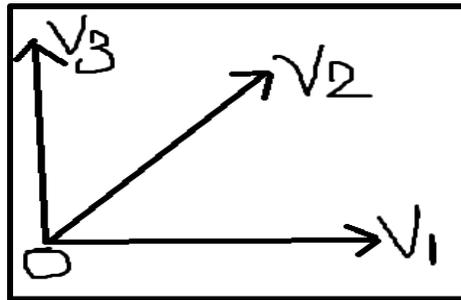


Fig: 7.27 Basics Vectors

As we have seen in stages in 3D viewing, different stages generate and work on different coordinate system, given as:

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

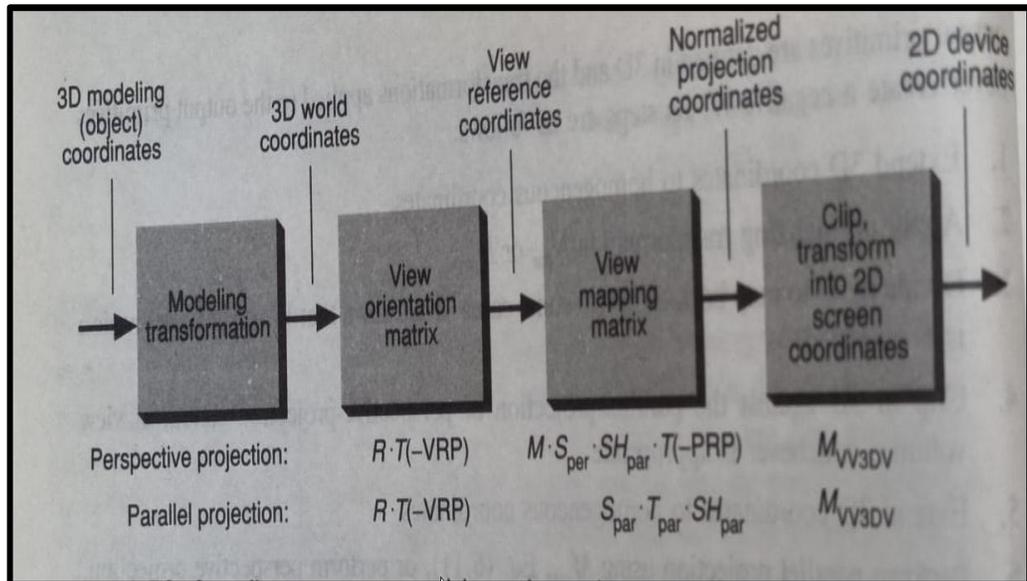


Fig: 7.28 Coordinate systems and how they relate to one another.

The matrices underneath each stage effect the transformation applied at 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 Fig. 7.28, individual objects are well-defined in an object-coordinate system, PHIGS (**P**rogrammer's **H**ierarchical **I**nteractive **G**raphics **S**ystem) 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 $[-1, 1]$, 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.

7.8 CAMERA MODEL AND VIEWING PYRAMID

- In 3D computer graphics, the view frustum (also called viewing frustum) 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 the adaption of (idealized) cone of vision that a camera or eye would have to the rectangular viewports typically used in computer graphics mentioned in below **figure: 7.29**.
- 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.

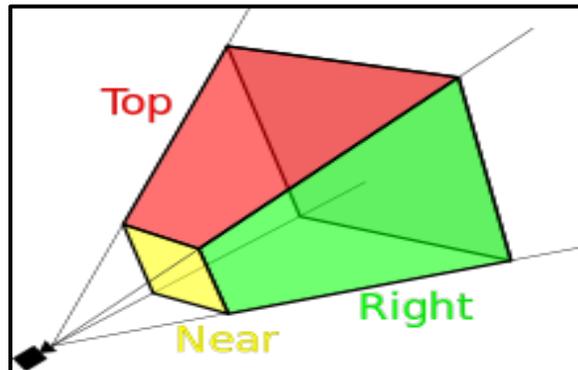


Fig: 7.29 View frustum

- 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 bounding volumes surrounding the objects rather than the objects themselves.

7.9 SUMMARY

- Viewing in 3D is divided mainly into following stages:
 - Modelling transformation
 - Viewing transformation
 - Projection transformation
 - Normalization transformation and Clipping
 - View Port transformation
- **Canonical View Volume (CVV):** Canonical, in computer graphics, is the standard state or Behavior of an attribute. This term is where it is used to refer to concepts that are unique and/or natural.

- Specifying an arbitrary view is meant to specify the following:
 - A View Reference Point (VRP) which is to point on the plane.
 - A View Plane Normal (VPN) which is the normal vector to the plane.
 - A View Up Vector (VUP) which is a vector from which we determine which way is up.
- Examples of 3D Viewing: In 3D viewing parameters, we set values for VRP, VPN, VUP, PRP, VRC and F and B of VRC based on the type of projection along with x, y and z principal axes. Where,
 - VRP - View reference point
 - VPN - View plane normal
 - VUP - View up vector
 - PRP - Projection Reference point
 - VRC - Viewing reference co-ordinate
 - Two-point perspective projection of the house
 - Two-point perspective projection of the house
 - Default perspective projection view volume- pyramid-like
- The Mathematics of Planar Geometric Projections: Planar projections are the subset of 3D graphical projections constructed by linearly mapping points in three-dimensional projection plane.
 - Mathematically, planar projections are linear transformations acting on a point in three-dimensional space $a(x, y, z)$ to give a point $b(u, v)$ on the projection plane.
 - Orthographic projection (known as orthogonal projection), is a means of representing three-dimensional objects in two-dimensions.
- Combined transformation matrices for projections and viewing: $MTOT = A_{wv} * M_p * N$
 - MTOT: - Represents Combined transformation matrices for projection and viewing
 - A_{wv} : - Represents Metrics of real-world coordinates
 - M_p : - Represents Metrics of perspective transformation
 - N : - Represents Normalization Matrix
- **Coordinate Systems and matrices:** Different stages generate and work on different coordinate system, given as:
 - Screen coordinates
 - Local co-ordinate system or 3D modelling coordinate
 - Word coordinate

- View reference coordinates
 - Normalized projection coordinates
 - 2D device coordinate
- **Camera model and viewing pyramid:** the view frustum (also called viewing frustum) 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”.

7.10 UNIT END EXERCISE

- Q.1 Explain the concept of viewing in 3D.
- Q.2 Explain with neat labelled diagram stages in 3D Viewing Pipeline with diagram.
- Q.3 Explain Canonical View Volume (CVV) in brief.
- Q.4 Explain a procedure which specifies an arbitrary 3D view.
- Q.5 Explain the way in which the different examples in 3D viewing is generated.
- Q.6 Briefly explain the mathematics of planar geometry.
- Q.7 Explain combined transformation matrices for projections and viewing.
- Q.8 Explain different coordinate systems and matrices in detail.
- Q.9 Explain the concept of camera model and viewing pyramid with neat labelled diagram.

7.11 LIST OF REFERENCES

- Computer Graphics - Principles and Practice by J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Publisher- Pearson, 2nd Edition
- Computer Graphics by A. P. Godse and D.A. Godse Technical Publication.
- Computer graphics with Virtual Reality Systems by Rajesh K. Maurya
- Computer Graphics by Hearn, Baker, Publisher-Pearson, 2nd Edition.
- Mathematical Elements for CG by D. F. Rogers, J. A. Adams, Publisher- TMH 2nd Edition.

LIGHT: - LIGHT: RADIOMETRY, TRANSPORT, EQUATION, PHOTOMETRY

Unit Structure

- 8.0 Objectives
- 8.1 What is Light?
- 8.2 Radiometry
 - 8.2.1 Radiant Energy
 - 8.2.2 Spectral Radiant Energy
 - 8.2.3 Radiant Flux (Radiant power)
 - 8.2.4 Spectral Radiant Flux
 - 8.2.5 Radiant Flux Density (irradiance and radiant Exitance)
 - 8.2.6 Spectral Radiant Flux Density
 - 8.2.7 Radiance
 - 8.2.8 Spectral Radiance
- 8.3 Transport
 - 8.3.1 Rasterization
 - 8.3.2 Ray Casting
 - 8.3.3 Ray Tracing
 - 8.3.4 Radiosity
- 8.4 Equation
- 8.5 Photometry
- 8.6 Summary
- 8.7 Unit End Exercise
- 8.8 List of References

8.0 OBJECTIVES

After going through this chapter, you will be able to:

- define light
- define and understand the concept of Radiometry and its different variants like Radiant Energy, Spectral Radiant Energy, Radiant Flux, Spectral Radiant Flux, Radiant Flux Density, Spectral Radiant Flux Density, Radiance, Spectral Radiance
- describe transportation of light and also understand its various

modeling techniques like ray tracing, rasterization, ray casting and radiosity.

- learn and understand the working of ray tracing technique
- state and derive transport equation of light
- describe the concept of Photometry

8.1 WHAT IS LIGHT?

- Light is defined as an electromagnetic radiation. The visible light that we see is only a tiny fraction of the electromagnetic spectrum, extending from very low frequency radio waves through microwaves, infrared, visible and ultraviolet light to x-rays and ultra-energetic gamma rays.
- Our eyes respond to visible light, identifying rest of the spectrum needs an arsenal of technical tools ranging from radio receivers to impressive counters.
- Light is radiant energy. When light is absorbed by a physical object, its energy is converted into some other form.
- **For example:** - In microwave oven, when we heats a glass of water then its microwave radiation is absorbed by the water molecules. The radiant energy of the microwaves is converted into thermal energy (heat).
- Similarly, visible light causes an electric current to flow in a photographic light meter then its radiant energy is transferred to the electrons as kinetic energy. Radiant energy (**denoted as Q**) is measured in **joules**.

8.2 RADIOMETRY

- Radiometry is the science of measuring radiant energy transfers. Radiometric quantities have physical meaning and can be directly measured using proper equipment such as spectral photometers.
- Radiometry is the science of measuring light in any portion of the electromagnetic spectrum. In practice, the term is typically limited to the measurement of infrared, visible and ultraviolet light using optical instruments. There are two aspects of radiometry: **theory and practice**.
- The practice includes the scientific instruments and materials used in measuring light, including radiation thermocouples, bolometers, photodiodes, photosensitive dyes and emulsions, vacuum phototubes, charged-coupled devices and a plethora of others.

❖ **Units of radiometry:**

Radiometric quantities are expressed in radiant units (such as watt) in

general case or in visual units when we are only interested in the fraction of light in the visible part of the spectrum. Visual units are useful in many applications such as multimedia, lightning system etc.

Table below shows radiometric quantities in Radiant and Visual units.

Quantities	Symbol	Radiant units	Visual Units
Energy	Q	Joule (J)	Joule(J)
Flux	F,P(ϕ)	Watt (W)	Lumen (lm)
Irradiance/Illuminance	E	W.m ⁻²	Lux
Intensity	I	W.sr ⁻¹	Candela (cd)
Radiance/Luminance	L	W.m ⁻² .sr ⁻¹	Cd.m ⁻²

Table 8.1

❖ **Uses of Radiometry:**

Following are the application areas based on Radiometry:

- **Public:** Camera, Photography, TV.
- **Biomedical:** Optical instrumentation, Medical imaginary.
- **Industry:** Photovoltaic, Lightning, Security, Non-destructive testing.
- **Spatial:** Planetary or deep space observation, Satellite design.
- **Defense:** Identification, Navigation.

8.2.1 Radiant Energy:

- Light is radiant energy. Electromagnetic radiation (wave and particle, depends on the measurement) transport energy through space. When light is absorbed by physical objects, its energy is converted into some other form. **For example**, Microwave oven heats a glass of water then its microwave radiation is absorbed by the water molecules. The radiant energy of the microwaves is converted into thermal energy (heat).
- Similarly, visible light causes an electric current to flow in a photographic light meter when its radiant energy is transferred to the electrons as kinetic energy.
- Radiant energy is denoted as **Q** and is measured in joules.

8.2.2 Spectral Radiant Energy :

- A broadband source such as the sun emits electromagnetic radiation throughout most of the electromagnetic spectrum, from radio waves to gamma rays. Though, most of its radiant energy is connected within the visible portion of the spectrum. On the other hand, a single wavelength laser is a monochromatic source; all of its radiant energy is emitted at one specific wavelength.
- Spectral radiant energy is the amount of radiant energy per unit wavelength interval at wavelength λ . It is given by,

$$\text{Spectral Radiant energy} = \text{Radiant energy} / \text{wavelength}$$

Defined as, $Q_\lambda = dQ/d\lambda \dots (8.1)$

- Spectral radiant energy is measured in **joules per nanometer**.

8.2.3 Radiant Flux (Radiant Power):

- Energy per unit time is power, which we measure in joules per seconds or watts.
- Light flows through space, and so radiant power is more commonly referred to as the time rate of flow of radiant energy, or radiant flux which is denoted by ϕ .
- It is given by, **Radiant flux (power) = Radiant energy/time**

Defined as,

$$\phi = dQ/ dt \dots(8.2) \text{Where,}$$

Q is radiant energy and t is time.

8.2.4 Spectral Radiant Flux:

- Spectral radiant flux is radiant flux per unit wavelength interval at **wavelength λ** .
- Spectral radiant flux is denoted by ϕ_λ , it is given by,
- Spectral Radiant flux = **radiant flux/ wavelength**

It is measured in watts per nanometer.

8.2.5 Radiant Flux Density (Irradiance And Radiant Exitance):

- Radiant flux density is the radiant flux per unit area at a point on a surface, where the surface can be real or imaginary (**i.e.** an exact plane).
- **There are two possible conditions:**

Irradiance: The flux can be arriving at the surface, in which case the radiant flux density is referred to as irradiance. The flux can arrive from any direction above the surface, as indicated by the rays.

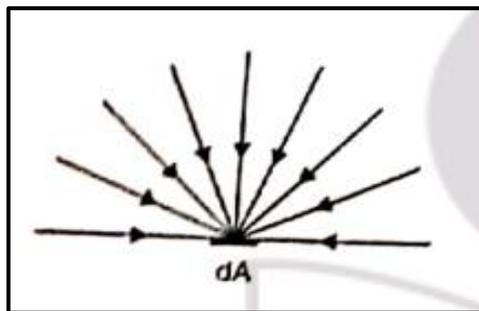


Fig: 8.1 Irradiance

- Irradiance is defined as: $E = d\phi/dA$

Where,

ϕ is the radiant flux arriving at the point.

dA is differential area surrounding the point.

Radiant exitance: The flux can be leaving the surface due to emission

and/or reflection (**fig 8.2**). The radiant flux density is then referred to as radiant exitance.

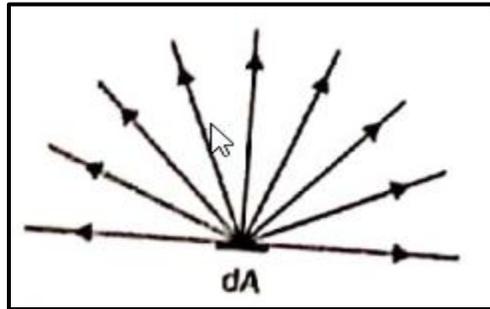


Fig: 8.2 Radiance Exitance

- As with irradiance, the flux can leave in any direction above the surface.
- The definition of radiant exitance is: $M = d\phi/dA$

Where,

ϕ Is the radiant flux leaving the point.

dA is the differential area surrounding the point

- Radiant flux density can be measured anywhere in three dimensional space between them (e.g. in air or a vacuum), and inside transparent media such as water and glass.
- Radiant flux density is measured in **watts per square meter**.

8.2.6 Spectral Radiant Flux Density :

- Spectral radiant flux density is radiant flux per unit wavelength interval at wavelength λ .
- When the radiant flux is arriving at the surface, it is called spectral irradiance denoted by E_λ , and is defined as:

$$E_\lambda = dE/d\lambda \dots\dots (8.3)$$

- When the radiant flux is leaving the surface, it is called spectral radiant exitance denoted by M_λ , and is defined as:

$$M_\lambda = dM/d\lambda \dots\dots (8.4)$$

- Spectral radiant flux density is measured in **watts per square meter per nanometer**.

8.2.7 Radiance:

- Radiance is the radiant flux emitted, reflected, transmitted or received by a given surface, per unit solid angle per unit projected area.
- Where, a solid angle is the 2D angle in 3D space that an object subtends at a point. It is a measure of how large the object appears to an observer looking from that point. In the international system of units (SI), a solid angle is expressed in a dimensionless unit called a *steradian*.

- A more formal definition of radiance requires that we think of a ray as being an extremely small narrow (“elemental”) cone with its apex at a point on a real or imaginary surface. This cone has a differential solid angle $d\omega$ that is measured in steradian.
- We must also note that the ray is intersecting the surface at an angle. If the area of intersection with the surface has a differential cross-sectional area dA , the cross-sectional area of the ray is $dA\cos\theta$, where θ is the angle between the ray and the surface normal, as shown in (fig: 8.3) where, the ray cross-sectional area $dA\cos\theta$, is called the projected area of the ray-surface intersection area dA .
- With these preliminaries in mind, we can imagine an elemental cone $d\omega$ containing a ray of light that is arriving at or leaving a surface (fig. 8.4 and 8.5). the definition of radiances is then:

$$L = d^2\phi/[dA(d\omega\cos\theta)] \dots\dots\dots(8.5)$$

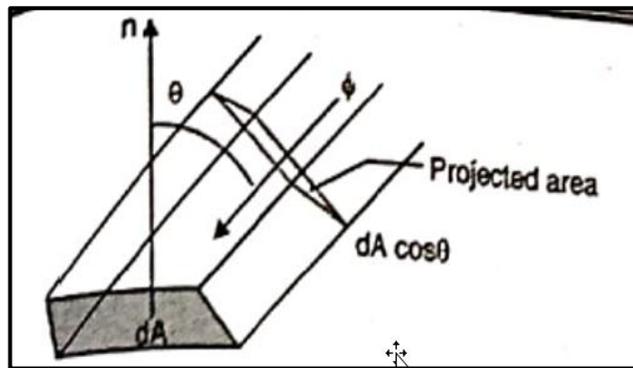


Fig. 8.3: Ray of light intersecting a Surface

Where, ϕ is the radiant flux,

dA is the differential area surrounding the point,

$d\omega$ is the differential solid angle of the elemental cone,

θ is the angle between the ray and the surface normal n at the point.

- Unlike radiant flux density, the definition of radiance does not distinguish between flux arriving at or leaving a surface.

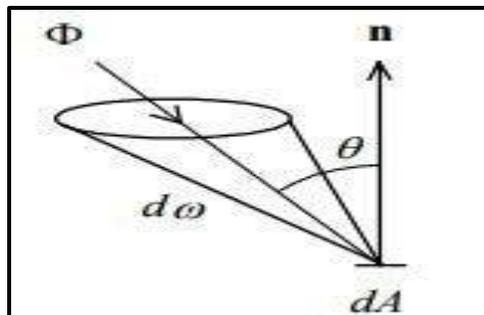


Fig. 8.4 Radiance (arriving)

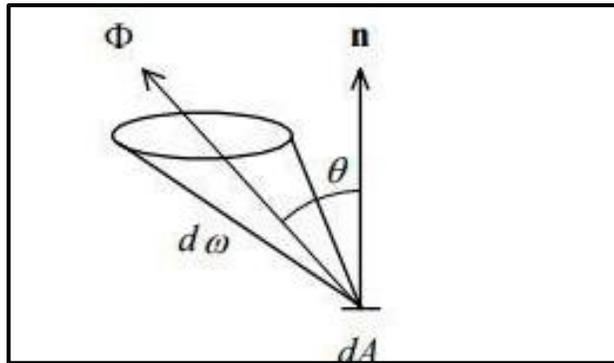


Fig. 8.5 Radiance (leaving)

- Another way of looking at radiance is to note that the radiant flux density at a point on a surface due to single ray of light arriving (or leaving) at an angle θ to the surface normal is, $d\phi/(dA\cos\theta)$.
- The radiance at that point for the same angle is then $d^2\phi/[dA(d\omega\cos\theta)]$, or radiant flux density per unit solid angle.
- Radiance is measured in watts per square meter per steradian.

8.2.8 Spectral Radiance:

- Spectral radiance is radiance per unit wavelength interval at wavelength λ denoted by L_λ . It is defined as:

$$L_\lambda = d^3\phi/[dA(d\omega\cos\theta)d\lambda] \dots (8.6)$$

And is measured in watts per square meter per steradian **per nanometer**.

8.3 TRANSPORT

- The amount of light transported is measured by flux density, or luminous flux per unit area on the point of the surface at which it is measured.
- Light transport theory deals with the mathematics behind calculating the energy transfers between media that affect visibility.
- Rendering converts a model into an image either by simulating a method such as light transport to get physically based photorealistic images, or by applying some kind of style as non-photorealistic rendering.
- The two basic operations in light transport are transport (how much light gets from one place to another) and scattering (how surfaces interact with light).
- Many rendering algorithms have been researched and software used for rendering are employed by using a number of different techniques to obtain a final image.
- Tracing every particle of light in a scene is nearly always completely

impractical and takes more amount of time.

- Even tracing a large enough portion produce an image that takes an excessive amount of time if the sampling is not intelligently limited.

Therefore a few families of more-efficient light transport modeling techniques have emerged and they are as follows:

8.3.1 Rasterization :

- It assumes an image defined in a vector graphics format (shapes) and converting it into a raster image (a series of pixels, dots or lines, which, when displayed together, creates image which was represented via shapes).

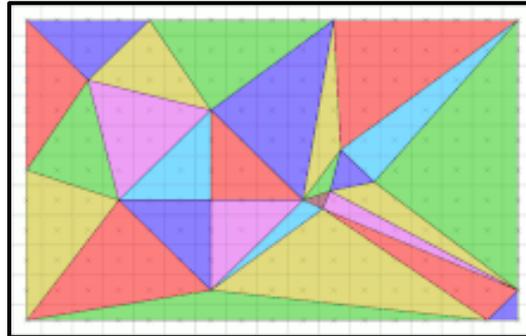


Fig: 8.6 Rasterization

- The rasterized image may then be displayed on a computer display, video display or printer, or stored in a bitmap file format.
- Rasterization may refer to either the conversion of models into raster files, or the conversion of 2D rendering primitives such as polygons or line segments into a rasterized format.

8.3.2 Ray Casting:-

- In 1982 this term was first stated in computer graphics by Scott Roth to define a technique for rendering constructive solid geometry models.

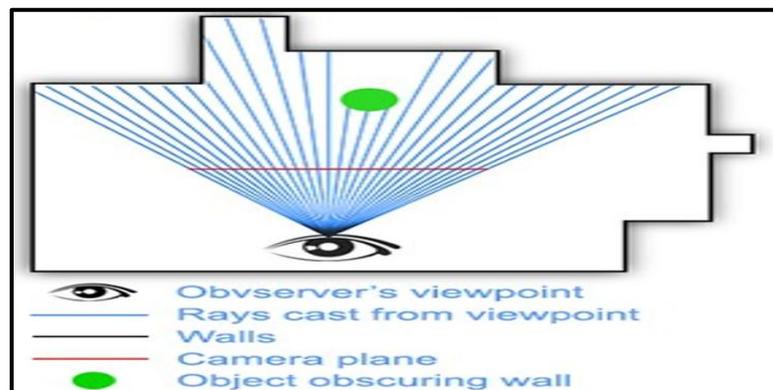


Fig: 8.7 Ray casting- Scenario

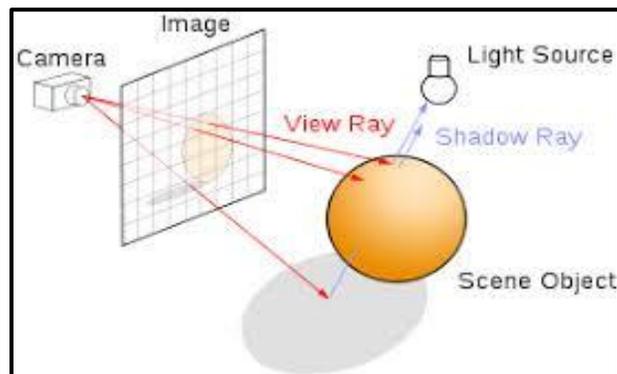
- It is a technique that alters a limited form of data into a 3D-projection by tracing the ray from view-point into the viewing volume.

- Ray casting is much more simplified way than ray tracing and it can be done in real time applications.
- Ray casting calculates a color for each pixel in the image plane by firing a ray through the view volume.
- It considers the scene as observed from a specific point of view, calculating the observed image based only on geometry and very basic optical laws of reflection intensity, and perhaps using **Monte Carlo technique** to reduce artifacts.

8.3.3 Ray tracing:

- This is similar to ray casting, but employs more progressive optical simulation, and uses Monte Carlo techniques to obtain more realistic results.
- It is a technique for generating an image by tracing the path of light through pixels in an image plane.
- They apply scan-line rendering method. But there is greater computational cost involved.
- Ray tracing is mostly used for application where image can be rendered slowly ahead of time such as in still image, film, T.V and poorly used for application like computer games where speed is critical.

❖ How does ray tracing works?



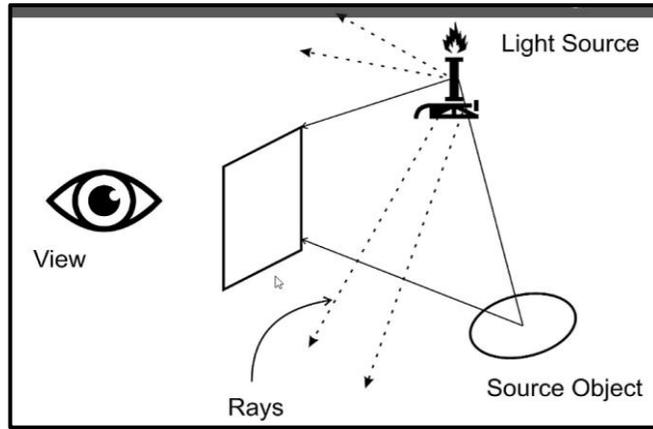


Fig. 8.8 Tracing ray from light sources to eye

- a) A ray tracing is so named because it tries to stimulate the path that light rays take as they bounce around within the world.
- b) They are outlined through the scene.
- c) The objective is to determine the color of each light ray that strikes the view window before reaching the eye.
- d) In this method consider tracing one ray through a scene with one object as mention in above **figure: 8.8**.
- e) We start from light bulb source then we need to decide, how many rays to shoot out from the bulb, then for each ray we have to decide in what direction it is going.
- f) There is infinity of directions in which it can travel, to choose one direction we are tracing a number of photons, some will reach the eye directly, other will bounce around, some and other will reach the eye and many more will never hit the eye at all.
- g) For all the rays that will never reach the eye, effort for producing tracing will be of no use.
- h) In order to save our wasted effort, we need to trace only those rays that are guaranteed to hit the view window and reach the eye.
- i) Instead of tracing rays starting at light source we trace them backward starting at the eye.
- j) Consider any point on view window whose color we are trying to determine. Its color depends on the color of light rays that pass through that point in view window and reach the eye.
- k) For solving this problem we develop **two rays: Original ray and backward ray**.
 - i) If Original ray come directly from light source, then backward ray will go directly to light source.
 - ii) If Original ray bounce off the table first, then the backward ray will also bounce off the table.
 - iii) We can clearly see this looking at the above figure again by reversing the directions of the arrow.

- iv) So backward method does same thing as the original method but it will not waste any kind of efforts for the rays that will never reach the eye.

8.3.4 Radiosity:

- The radiosity method of light interaction was developed by researchers at Cornell University and Hiroshima University, as a method for computing radiant heat exchange between surfaces.
- This is a method which attempts to pretend the way in which directly illuminated surfaces act as indirect light sources that illuminate other surfaces. This produces more realistic shading and seems to better capture the ‘ambience’ of an indoor scene. A classic example is the way that shows ‘hug’ the corners of rooms.
- It is a global illumination algorithm used in 3D computer graphics rendering.
- It is an application of the finite element method to solve the rendering equation for scenes with diffused surface.

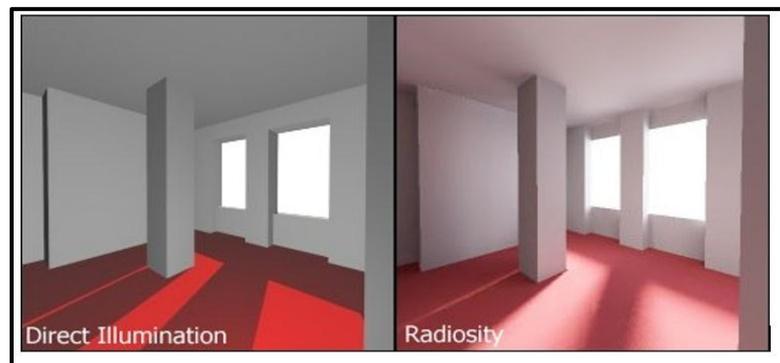


Fig. 8.9 Radiosity

- The image on the left was rendered with a typical *direct illumination renderer*. There are three types of lighting in this scene which have been exactly chosen and placed by the artist in an attempt to create realistic lighting:
 - *Spot lighting* with shadows (placed outside the window to create the light shining on the floor)
 - *Ambient lighting* (without which any part of the room not struck directly by a light source would be totally dark)
 - *Omnidirectional lighting* without shadows (to reduce the flatness of the ambient lighting).
- The image on the right was reduced using a *radiosity algorithm*. There is only one source of light: an image of the sky placed outside the window. The difference is marked. The room glows with light. Soft shadows are visible on the floor, and indirect lighting effects are visible around the room. Besides, the red color from the carpet has drained onto the grey walls, giving them a somewhat warm advent. None of these effects were exactly chosen or planned by the artist.

- Because of this, radiosity is a prime component of leading real-time rendering methods, and has been used from beginning-to-end to create a large number of well-known recent feature-length animated 3D-cartoon films.
- **There are two types of radiosity:**
 - ❖ **Progressive radiosity:** - It solves the system iteratively in such a way that each iteration, we can get an intermediate radiosity value for the patch.
 - ❖ **Shooting radiosity:** - It iteratively solves the radiosity equation by shooting the light from the patch with most error at each step.

8.4 TRANSPORT: EQUATION

The light performance onto an object is analyzed and divided into three unique components. To actually compute the rays is too expensive to do in real-time.

Light at a pixel from a light = Ambient + Diffuse + Specular contributor

$$I_{light} = I_{ambient} + I_{diffuse} + I_{specular}$$

(As given in below diagram)

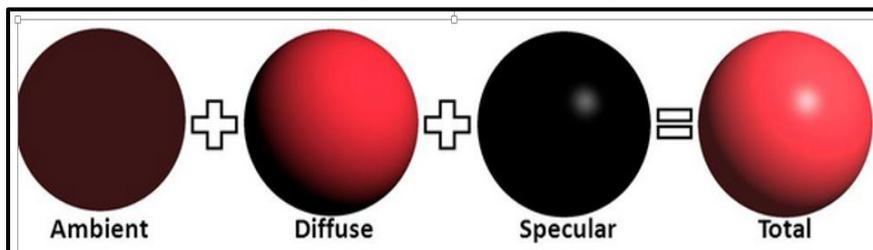


Fig. 8.10: Light Model Depending on Various Components

Most important equation for radiance can be obtained as follows:

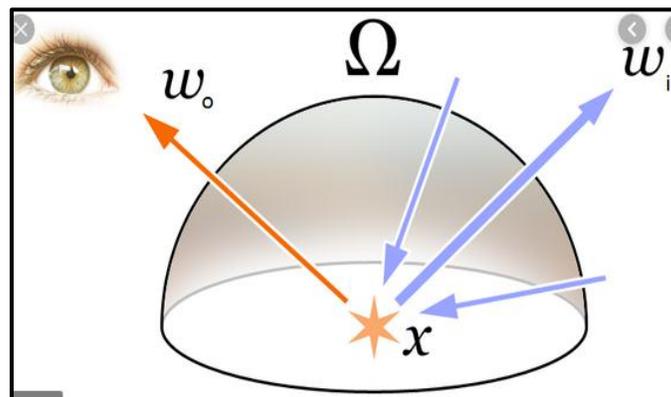


Fig. 8.11: Demonstration to calculate total radiance

Total radiance= emitted + reflected radiance

$$L(x, \omega_0) = L_e(x, \omega_0) + \int_{\Omega^+} fr(\omega_i, x, \omega_0) L_i(x, \omega_i) \cos\theta_i d\omega_i$$

In above equation we have,

x	Surface position
ω_0	Outgoing direction
ω_i	Incoming illumination direction
$fr(\omega_i x, \omega_o)$	BRDF: bidirectional reflectance distribution function
$L(x, \omega_0)$	Represent total visible surface radiance

Table: 8.2

8.5 PHOTOMETRY

- Photometry is the science of measuring visible light in units that are biased according to the sensitivity of the human eye.
- It is different from radiometry, which is the science of measurement of radiant energy (including light) in terms of absolute power.
- Photometry is a branch of science concerning light in terms of color apparent by the viewer from the physical inspiration of imposing photons into the eye and the combined response with the brain.
- The purpose of photometry is to measure light in a way that takes the sensitivity of human visual system into account. Photometry is essential for evaluation of light sources and objects used for lighting, signaling, displays, and other applications where light is envisioned to be seen by humans.
- It is a quantitative science based on a statistical model of the human visual responses to light that is, our perception of light under wisely precise conditions.
- The human visual system is an amazingly complex and highly nonlinear sensor of electromagnetic radiation with wavelength ranging from 380 to 770 nanometers (nm).
- Light of different wavelengths is perceived as a range of colors ranging through the visible spectrum **650 nm is red, 540 nm is green, and 450 nm is blue**, and so on.
- This task is complicated immensely by the eyes nonlinear response to light. It varies not only with wavelength but also with the amount of radiant flux, whether the light is constant or flickering, the spatial complexity of the scene being perceived, the alteration of the iris and retina, the psychological and physiological state of the observer, and a host of other variables.

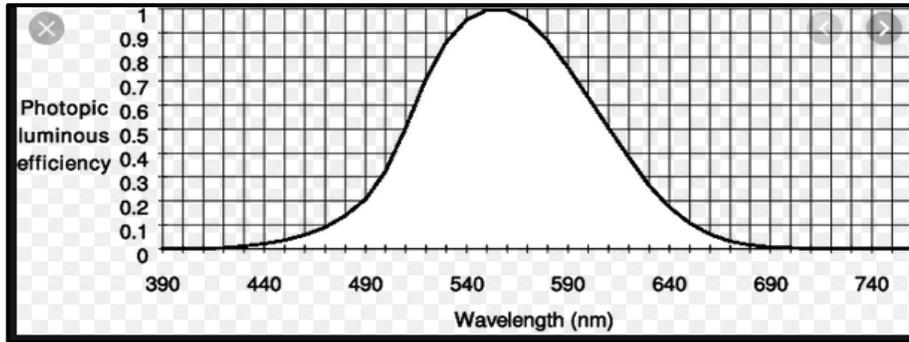


Fig. 8.12: CIE photometric Curve

- The Subjective impression of seeing can be quantified for “normal” viewing conditions. In 1924, the commission Internationale d’Eclairage (International Commission on Illumination, or CIE) enquired over one hundred observers to visually match the “brightness” of monochromatic light sources with different wavelengths under controlled conditions.
- The statistical result so called CIE photometric curve in above figure shows the photonic luminous effectiveness of the human visual system as a function of wavelength. It provides a weighting function that can be used to convert radiometric into photometric measurements.
- Photometric theory does not address how we notice colors. The light being measured can be monochromatic or a combination or continuum of wavelengths; the eye’s response is determined by the CIE weighting function. This emphasizes a vital point: The only difference between radiometric and photometric theory is in their units of measurement.

8.6 SUMMARY

- Light is electromagnetic radiation.
 - Electromagnetic radiation (which can be considered both wave and a particle, depending on how you it transports energy through space.
 - Radiant energy (denoted as Q) is measured in joules.
- Radiometry is the science of measuring radiant energy transfers. There are two aspects of radiometry theory and practice.
 - Units of radiometry quantities are expressed in radiant units (such as watt) in general case or in visual units when we are only interested in the fraction of light in the visible part of the spectrum.
 - Spectral radiant energy is measured in joules per nanometer.
 - Radiant Flux (Radiant power): Q is radiant energy and t is time.
 - Spectral radiant flux is radiant flux per unit wavelength interval at wavelength λ .

- Radiant Flux Density (irradiance and radiant Exitance)
- Spectral radiant flux density is radiant flux per unit wavelength interval at wavelength λ .
- Spectral radiance is radiance per unit wavelength interval at wavelength λ denoted by L_λ .
- **Light Equation:** Light at a pixel from a light = **Ambient + Diffuse + Specular contributor.**
- A **ray tracing** is a technique for generating an image by tracing the path of light through pixels in an image plane.
- A **ray casting** is a much more simplified version of the ray tracing and can be easily done in real time.
- **Radiosity** is an application of the finite element method to solve the rendering equation for scenes with purely diffuse surfaces.
- Photometry is the science of measuring visible light in units that are weighted according to the sensitivity of the human eye. We see light of different wavelengths as a continuum of colors ranging through the visible spectrum: 650 nm is red, 540 nm is green, and 450 nm is blue, and so on.

8.7 UNIT END EXERCISES

Q.1 Explain Radiometry in brief.

Q.2 Explain the following terms:

- a) Radiant energy
- b) Spectral Radiant energy
- c) Radiant Flux or Radiant power
- d) Spectral Radiant Flux or Radiant power
- e) Radiant Flux Density
- f) Irradiance radiant exitance
- g) Spectral Radiant Flux density
- h) Radiance

Q.3 Explain units and uses of Radiometry.

Q.4 What is meant by radiance and irradiance?

Q.5 Write a short note on rasterization.

Q.6 Explain ray casting technique with diagram in detail.

Q.7 How does ray tracing works? Explain in detail.

Q.8 Write a short note on radiosity.

Q.9 Write a short note on Transport equation.

Q.10 Explain the concept of photometry in detail.

8.8 LIST OF REFERENCES

- Computer Graphics by A. P. Godse and D.A. Godse Technical Publication.
- Computer graphics with Virtual Reality Systems by Rajesh K. Maurya
- Computer Graphics by Hearn, Baker, Publisher-Pearson, 2nd Edition.
- Computer Graphics - Principles and Practice by J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Publisher- Pearson, 2nd Edition.
- Mathematical Elements for CG by D. F. Rogers, J. A. Adams, Publisher- TMH 2nd Edition.

COLOR: - COLORIMETRY, COLOR SPACES, CHROMATIC ADAPTATION, COLOR APPEARANCE

Unit Structure

- 9.0 Objectives
- 9.1 What is Color?
- 9.2 Colorimetry
 - 9.2.1 Colorimetry Concepts in Display Test and Measurement
- 9.3 Color spaces
 - 9.3.1 RGB Color Model
 - 9.3.2 CMYK Color Model
 - 9.3.3 HSV Color Model
 - 9.3.4 HSL Color Model
- 9.4 Chromatic Adaptation
- 9.5 Color Appearance
- 9.6 Summary
- 9.7 Unit End Exercises
- 9.8 List of References

9.0 OBJECTIVES

After studying this Chapter, you will be able to:

- Understand the definition of Color, Colorimetry and different applications of it.
- Learn about Colorimetry Concepts in Display Test and Measurement.
- Know and define the different types of Color spaces, Color Wheel and Color Model.
- Understand chromatic adaptation and its various forms.
- Learn and state Color Appearance

9.1 WHAT IS COLOR?

- It is defined as an quality of an object or substance with respects to light reflected by an object, usually determined visually by measurement of hue, saturation and brightness of the reflected of light.

- It is also defined as an quality of visual insight consisting of any arrangement of chromatic and achromatic content this attribute can be described by chromatic color names such as yellow, orange, brown, red, pink, green, blue, purple, etc., or by achromatic color names such as white, gray, black, etc., and qualified by bright dim, light, dark, etc., or by combination of such names.
- Light is a mixture of radiations having different wavelength where different wavelength implies different colors.
- Light maybe decomposed into a spectrum that is the separation of photos (light quanta) with respects to the different wavelength i.e. different colors. **Illustrated in fig: 9.1**
- Spectrum is measured as Watt per square meter per nanometer. The spectrum is condition that has infinite numbers of independent elementary colors present in it.

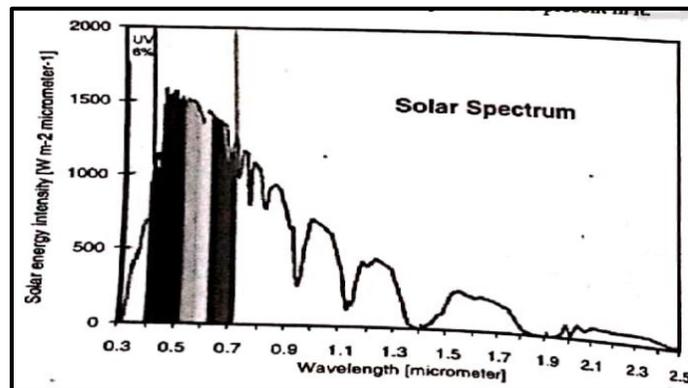


Figure 9.1 Spectrum of Light

9.2 COLOR: COLORIMETRY

- Science related to the perception of colors is known as Colorimetry.
- The Science of Colorimetry is used to quantify the response of the human visual system and match human color perception for applications in a variety of industries.
 - **Display Manufacturing:** - Quality control for industrial production lines and incoming inspection of display glass. Display calibration for LED, LCD, Plasma, projection, CRT displays.
 - **Broadcasting:** - Measuring and standardizing video walls for color accuracy, uniformity of brightness and white balance.
 - **Graphic Design and Computer Animation:** - Professionals who rely on color accuracy and precision color measurement benefit from understanding Colorimetry.

9.2.1 COLORIMETRY CONCEPTS IN DISPLAY TEST AND MEASUREMENT:

- **Color:** Color perception is naturally independent. The objective capacities of the color of a source object is alleged by a standard human observer that also can be measured. Various standard human observers are defined in the discipline of Colorimetry. According to standard model, the perceived color of a given spot can be reduced to a three-dimensional value. The three dimensions of color can be described as color of brightness, hue and purity or saturation.
- **Chromaticity:** A two-dimensional explanation of color has the mixture of hue and purity, neglecting the third dimension of brightness. The luminance (brightness) and chromaticity of a spot on a display, taken together, provide a complete depiction of its color.
- **Gamut mapping (RGB plotting):** All colors that are produced by a display device are created by combination of Red, Green and Blue colors. Each display color can be described in terms of the amount of R, G and B primaries present. If the chromaticity directs the three primaries are plotted in a chromaticity diagram, the triangle enclosed by these points represents the full range of colors reproducible by the display. This range is the displays of color gamut.
- **Correlated Color Temperature (CCT):** This is a metric used to define the color of a white light by comparing its chromaticity to that of an idealized radiant source, known as a black body. The color of a bright source (which glows due to heat) depends upon its temperature; lower temperature sources are bluer. The CCT of a white light is the high temperature of the black body which almost closely matches its chromaticity.
- **White Balance, Gray Balance:** One of the vital features of color depiction is the position of the white point. The chromaticity of a specified white light, to which the observer is presumed to be improved. Light sources with different CCT have slightly different chromaticity's, but once the location of the desired white point is specified, the proportion of R, G and B primaries required to reproduce are classically non-linear, a different gray balance may be required to reproduce the same chromaticity at a lower luminance level.
- **Dominant Wavelength and Purity:** These values taken together to represent a substitute description of chromaticity. Dominant wavelength corresponds to hue, while purity corresponds to saturation. The relationship between wavelength and hue can be tacit in terms of the colors of the visible spectrum, as perceived in the rainbow: Shorter wavelengths resemble to violet and blue hues; medium wavelengths to greens and yellow; longer wavelengths to orange and red hues.

9.3 COLOR SPACES

- A range of colors that can be created by the primary colors of pigment and these colors then define a specific color space.
- It is a way to represent colors, usually used in relation to computers or graphics boards. Color includes levels of grey. Number of bits specifies color space. 1 bit color is black and white combination. Grey scale pattern refers 2 bits to generate black, dark grey, light grey, white colors.
- Color space, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g. RGB). It is an elaboration of the co-ordinate system and sub-space. Each color in the system is represented by a single dot.
- A **Color wheel** is a tool that offers a visual representation and relationships between all possible hues.
- The primary colors are arranged around a circle at equal (120 degree) intervals. Color Wheels often depict “Painter’s Colors” primary colors, which leads to a different set of hues.

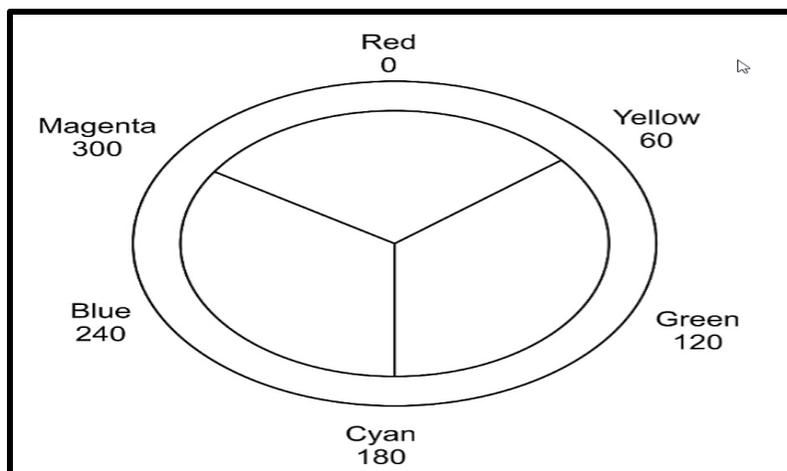


Fig 9.2: Color Wheel

- A **Color model** is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called color space. This section describes ways in which human color vision can be modeled.
- There are three different color models that describes the different perceived characteristics of color are **RGB model**, **CMY model**, **HSV or HLS Model**.

9.3.1 RGB Color model:

- The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.
- The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in a conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based on the human perception of colors.

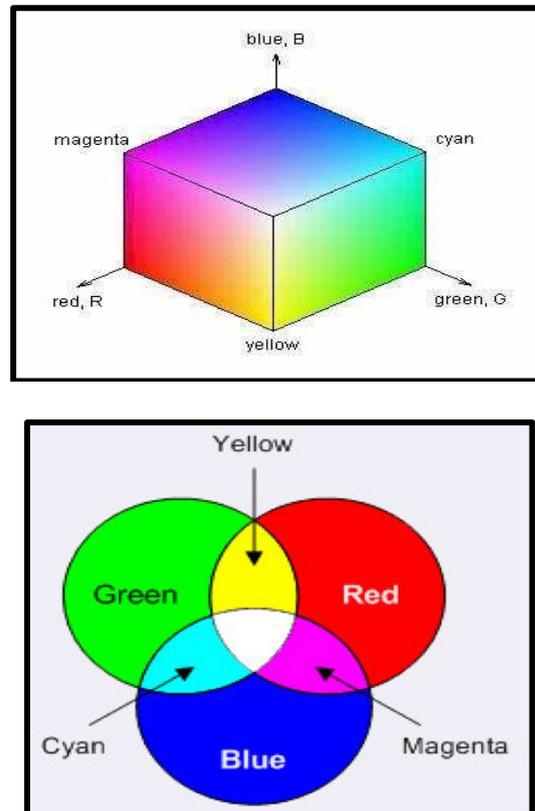


Fig 9.3: RGB color model

- In the RGB color model, we use red, green, and blue as the three primary colors. We do not actually specify what wavelengths these primary colors correspond to, so this will be different for the different types of output media, for example, different monitors, films, videotapes, and slides.
- This is an additive model since the phosphors are emitting light. A subtractive model would be the one in which the color is the reflected light. We can represent the RGB model by using a unit cube. Each point in the cube (or the vector where the other point is the origin) represents a specific color. This model is the best for setting the electron guns for a CRT (**refer Figure: 9.3**)

9.3.2 CMY Color model:

- CRTs produce color by the emission and use of the RGB model. Printers produce color by the reflective light, so it is a subtractive process and uses a model based on the colors i.e. cyan, magenta, and yellow (CMY).
- CMYK (short for cyan, magenta, yellow and key or black), or the process color (the four-color model) is a subtractive color model used in the color printing, also used to describe the printing process itself.
- The CMYK model works by partially or entirely masking, certain colors on the typically white background (that is absorbing particular wavelengths of light). Such a model is called subtractive because the inks subtract the brightness from white (see **Figure 9.4**)

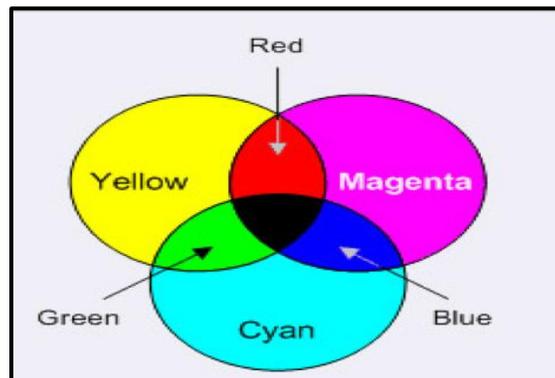


Fig 9.4: CMY color model

- Remember that cyan-green+ blue, go the light reflected from a cyan pigment has no red component, that is, red is absorbed by cyan. Similarly, magenta subtracts green, and yellow subtracts blue.
- Printers usually use four colors: cyan, yellow, magenta, and black. This is because cyan, yellow, and magenta together produce a dark grey rather than a true black. The conversion between the RGB and CMY is easily computed as
- Cyan 1-Red
Magenta-1-Green
Yellow 1-Blue
- The reverse mapping is similarly, obvious.
Red-1-Cyan
Green-1-Magenta
Blue-1-Yellow

9.3.3 HSL Color model:

- An HSL model is based on the parameters **hue, saturation, and lightness**.
- The HSL is another way to describe a color with the three parameters. The RGB model is the way computer screens a work, but not very intuitive.

- The nicest application of this color model is that you can easily create the rainbow gradients or change the color, lightness or saturation of an image with this color model.

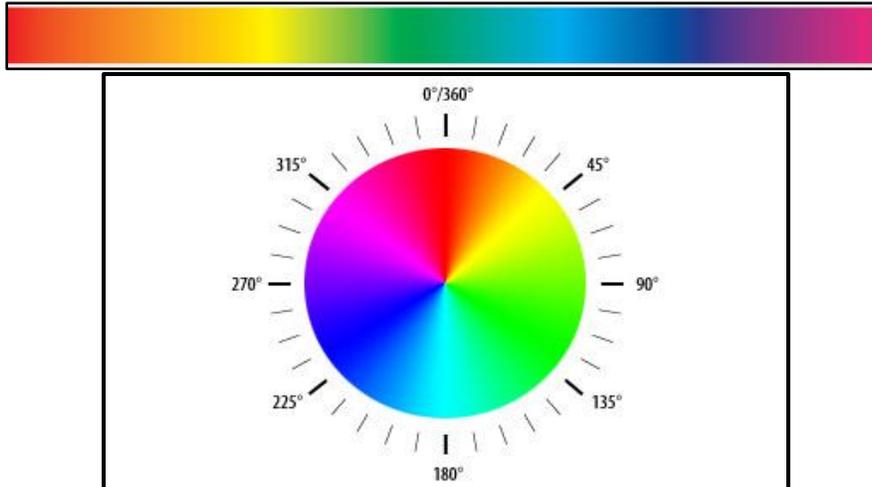


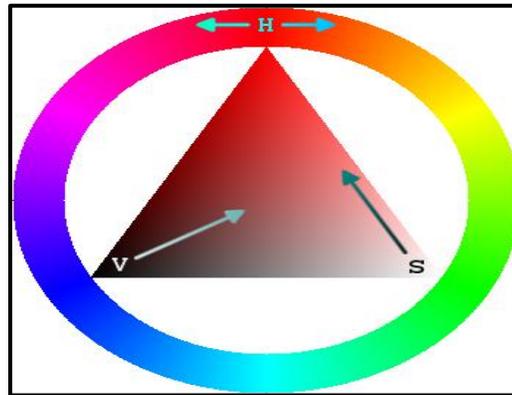
Fig 9.5: Hue in HSL Color model

- The hue indicates the color sensation of light. In other words, if the color is red, yellow, green, cyan, blue, and magenta, this representation looks almost the same as the visible spectrum of light, except on the right is now the color magenta (the combination of red and blue), instead of violet (light with a frequency higher than blue).
- The hue works circular, so it can be represented on a circle instead. A hue of 360° looks the same again as a hue of 0° . The saturation indicates the degree to which the hue differs from a neutral gray.
- The values run from 0 percent (which is no color) to 100 percent (which is the fullest saturation) of a given hue at a given percentage of the illumination. The more the spectrum of the light is concentrated around one wavelength, the more saturated the color will be
- The lightness indicates the illumination of the color. At 0 percent, the color is completely black, at 50 percent, the color is pure, and at 100 percent, it becomes white (see **Figure 9.5**).
- The HSL, color model is, **for example**, used in Corel Paint Shop Pro* Photo X2 color picker and other color-processing software packages.

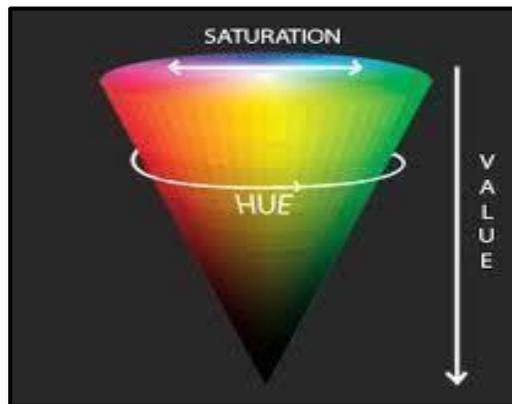
9.3.4 HSV Color model:

- The HSV (hue, saturation, value), also called the HSB (hue, saturation, brightness) model, defines a color space in terms of three constituent components: hue, saturation, and value.
- The HSV color model is more intuitive than the RGB color model.
- The user specifies a color (hue) and then adds white or black. Changing the saturation parameter corresponds to adding or subtracting white and changing the value parameter corresponds to adding or subtracting black.
- A 3D representation of the HSV model is derived from the RGB mode

cube. If we look at the RGB, the cube along the gray diagonal, we can see a hexagon that is the HSV hexcone. The hue is given by an angle about the vertical axis with red at 0° , yellow at 60° , green at 120° , cyan at 180° , blue at 240° , and magenta at 300° . Note that the complementary colors are 180° apart.



(a)



(b)

Fig 9.6: 3D presentation of an HSV model
(a) Color model (b) Color cone

- Saturation S , varies between 0 and 1, is the ratio of purity of a related hue to its maximum purity at saturation equal to 1. Also $S=0$ yields the grayscale corresponding to value V of HSV that is the diagonal of the RGB cube with $V=0$ being black and $V=1$ corresponding to white.
- The HSV color wheel (refer to **Figure 9.6(a)**) is used to pick the desired color. The hue is represented by a circle in the wheel. A separate triangle is used to represent the saturation and the value. The horizontal axis of the triangle indicates the value, and the vertical axis represents the saturation. When you need a particular color for your picture, first you need to pick a color from the hue (the circular region), and then from the vertical angle of the triangle, you can select the desired saturation. For brightness, you can select the desired value from the horizontal angle of the triangle.
- Sometimes, the HSV model is illustrated as a cylindrical or conical

object (**refer to Figure 9.6(b)**). When it is represented as a conical object, a hue is represented by a circular part of a cone. The cone is usually represented in a 3D form. A saturation is calculated using the radius of the cone, and the value is the height of the cone. A hexagonal cone can also be used to represent an HSV model. The advantage of the conical model is that it is able to represent the HSV color space in a single object. Due to the 2D nature of computer interfaces, the conical model of HSV is best suited for selecting colors for computer graphics.

- The colors used in HSV can be clearly defined by the human perception that is not always the case with RGB or CMYK.
- The HSV color space is widely used to generate high-quality computer graphics. In simple terms, it is used to select different colors needed for a particular picture. An HSV color wheel is used to select the desired color.
- A user can select a particular color needed for a picture from the color wheel. It gives the color according to the human perception.

9.4 CHROMATIC ADAPTATION

- The surroundings in which viewer objects and images has a larger effect on how we perceive those objects/ the range of viewing environment (i.e., by mean of light) is very large, from sunlight too moonlight or from candle light to luminous light. The lightning condition is not only involving a very high range of light but also varies greatly in the range of color which emits light.
- A human visual system accommodates these change in the environment through a process called as **adaptation**.

- There are **three types of adaptation**:

- Light adaptation
- Dark adaptation
- Chromatic adaptation

❖ **Light Adaptation:**

- 1) It refers to the change occurs when we move from very dark to very light environment i.e., dark -> light
- 2) When this happens we are dazzled at first by the light but soon we adapt to the new situation and then we begin to distinguish objects in our environment.

❖ **Dark Adaptation:**

- 1) It refers to the change occurs when we move from very light to very dark environment i.e., light -> dark
- 2) When this happens we see very little at first but after some time the details of an objects starts appearing in front of us.
- 3) Time needed to adapt objects in dark adaptation is much longer than

that of light adaptation.

❖ **Chromatic Adaptation:**

- 1) It refers to the human's ability to adjust and largely ignore differences in the color of the illumination. Although, we are able to largely ignore the changes in the viewing environment but we are unable to do it completely. For example, color appears much more colorful in a sunny day as compare to a cloudy day.
- 2) Chromatic adaptation is the ability of the human visual system to discount the color of a light source and to approximately preserve the appearance of an object. For example, a white piece of paper appears to be white when viewed under sky light and tungsten light (light under a light bulb). However, the measured tri-stimulus values are quite different for the two viewing conditions for e.g., sky light is bluer and contains shorter wavelength energy than tungsten light which is illustrated in **fig: 9.7.**

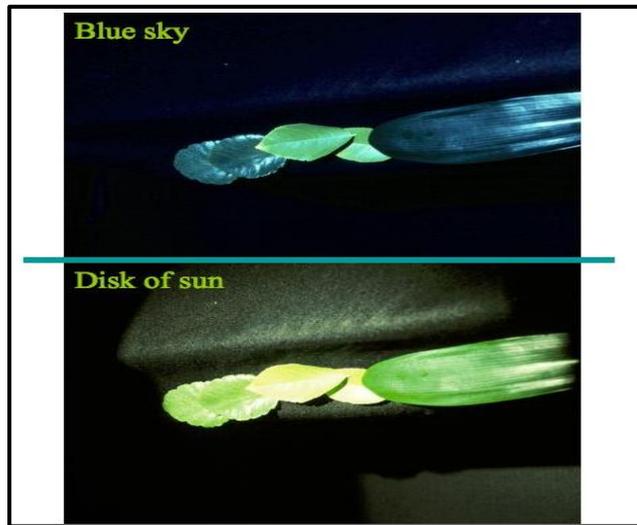


Fig 9.7: Chromatic adaptation under daylight and blue sky light

- 3) Chromatic adaptation is the biological equivalent of a white balancing operation that is available on most of the modern cameras. It allows white objects to appear white for a large number of lighting conditions.
- 4) Digital imaging systems, such as digital cameras and scanners, do not have the ability to adapt to the light source. Scanner usually use fluorescent light sources. For digital cameras the light source varies with the scene, and sometimes within a scene. Therefore, to achieve the same appearance of the original or original scene under different display conditions (such as a computer monitor or a light booth), the captured image tri-stimulus values have to be transformed to take into account the light source of the display viewing conditions such transformation are called chromatic adaptation transforms (CATs)
- 5) The adapting illumination can be measured off a white surface of a

scene. In a digital image, the adapting illumination can be approximated as the maximum tri-stimulus values of the scene.

9.5 COLOR APPEARANCE

- Colorimetry allows us to precisely require and communicate color in a device independent manner and chromatic adaptation allows us to predict color matches across changes in illumination but these tools are still inadequate to define how color actually look like.
- A Color appearance Model provides mathematical formulae to transform Physical measurements of the stimulus and viewing environment into Correlates of perceptual attributes of color (eg. **lightness, Chroma, hue, etc.**). Based on this colors are divided into four types, viz. **Nominal, Ordinal, Interval and Ratio**. Below **Fig 9.8** Shows this four types along with examples. These types are based on the values found after plotting as shown in the below **fig: 9.9**.

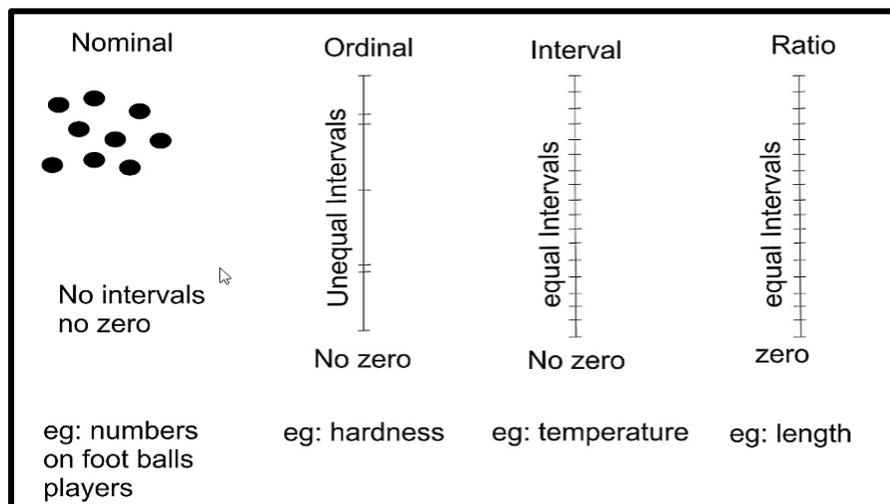


Fig 9.8: Stimulus Perceptual Attributes of Color

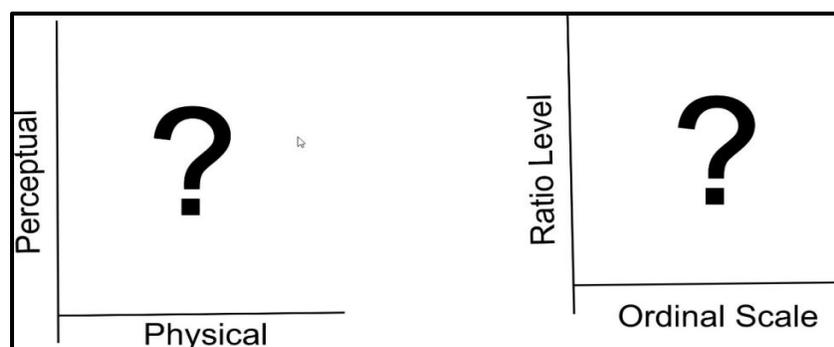


Fig 9.9: Plotting used for stimulus perceptual attributes of color

- To predict the actual perception of an object, we required to know more information about an environment. The human visual system is constantly adapting to this changing environment thus the perception of color is highly affected by such changes.

- There are different parameters used for color appearance which are given as follows:
 - HUE
 - Brightness
 - Lightness
 - Colorfulness
 - Chroma
 - Saturation
- **HUE:** It is an Attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors i.e. pure color: red, yellow, green, and blue, or to a combination of two of them. Hue is a more technical definition of our color perception which can be used to communicate color ideas.
- **Brightness:** It is an Attribute of a visual sensation according to which an area appears to emit more or less light. It is referred to as the absolute level of the perception.
- **Lightness:** It is a representation of variation in the perception of a color or color space's brightness. It is referred to as relative brightness normalized for changes in the illumination and viewing conditions. Lightness defines a range from dark (0%) to fully illuminate (100%). Any original hue has the average lightness level of 50%. Lightness is the range from fully shaded to fully tinted. We can lighten or darken a color by changing its lightness value
- **Colorfulness:** It is an attribute of a visual sensation according to which the perceived color of an area seems to be more or less chromatic (e. multiple color variations).
- **Chroma:** Chroma is a component of a color model. There's a blue yellow and a red-green chroma component.
- **Saturation:** Saturation is used to determine certain color and measured as percentage value. Saturation defines a range from pure color (100%) to gray (%) at a constant lightness level.
- A pure color is fully saturated. A de-saturated image is said to be dull, less colorful or washed out but can also make these attributes are also unable to predict the changes in impression of being softer.

A color appearance model (abbreviated CAM) is a mathematical model that seeks to describe the perceptual aspects of human color vision, i.e., viewing conditions under which the appearance of a color does not tally with the corresponding physical measurement of the stimulus source. (In contrast, a color model defines a coordinate space to describe colors, such as the RGB and CMYK color models.)

Thus there are several mathematical models are used to improve the color appearance.

Few color appearance models are listed below:

1. **CIELAB Model**- first color appearance model to focus on overall color improvement of an object
2. **Nayatani et al Model**- focuses on illumination engineering and the color rendering properties of light sources.
3. **Hunt Model**-focuses on color image reproduction
4. **RLAB Model**-focuses on color image reproduction
5. **LLAB Model**-focuses on color image reproduction
6. **CIECAM97A Model**-focuses on color image reproduction
7. **Model**-focuses on hue color formulation and suited for gamut mapping
8. **ICtCp Model**-improves the original IPT by exploring higher dynamic range and larger color gamut.

9.6 SUMMARY

- A **color model** is an abstract mathematical model that describes the way colors can be represented. It can be represented as tuples of numbers, typically as the three or four values of the color components. A CIE chromaticity diagram is an international standard for the primary colors established in 1931.
- Colorimetry is a science of color measurement and color appearance.
- **Colorimetry applications:**
 - Display Manufacturing
 - Broadcasting
 - Graphic Design and Computer Animation
- **Color space**, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g. RGB).
- **Chromatic adaptation** is the ability of the human visual system to discount the color of a light source and to approximately preserve the appearance of an object. There are three types of adaptation:
 - Light adaptation
 - Dark adaptation
 - Chromatic adaptation
- A **Color wheel** is a tool that provides a visual representation of the relationships between all possible hues.
- A **Color appearance Model** provides mathematical formulae to transform Physical measurements of the stimulus and viewing

environment into Correlates of perceptual attributes of color (eg. lightness, Chroma, hue, etc.)

9.7 UNIT END EXERCISES

- Q.1 What is color?
- Q.2 Explain Colorimetry and its different applications in brief.
- Q.3 Explain Colorimetry Concepts in Display Test and Measurement.
- Q.4 What is meant by Colorspaces and Color Wheel?
- Q.5 Explain following color model with diagram:
- a) RGB
 - b) CMY
 - c) HSV
 - d) HSL
- Q.6 Explain chromatic adaption in brief.
- Q.7 Explain Color appearance in brief.

9.8 LIST OF REFERENCES

- Computer Graphics by A. P. Godse and D.A. Godse Technical Publication.
- Computer graphics with Virtual Reality Systems by Rajesh K. Maurya.
- Computer Graphics by Hearn, Baker, Publisher-Pearson, 2nd Edition.
- Computer Graphics - Principles and Practice by J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Publisher- Pearson, 2nd Edition
- Principles of Interactive Computer Graphics by William M. Newman and Robert F. Sproull, Publisher- TMH, 2nd Edition.

VISIBLE-SURFACE DETERMINATION AND PLANE CURVES AND SURFACES

Unit Structure

10.0 Objectives

10.1 Techniques for efficient Visible-Surface Algorithms,

10.2 Categories of algorithms

10.2.1 Back face removal

10.2.2 Scan-line method

10.2.3 The z-Buffer Algorithm

10.2.4 Area sub-division method (Warnock's Algorithm)

10.2.5 BSP trees

10.2.6 Painter's algorithms (depth sorting)

10.2.7 Visible-Surface Ray Tracing

10.3 Comparison of the methods

10.4 Summary

10.5 Questions

10.6 References

References

10.0 OBJECTIVES

1. To know the meaning of Visible-surface detection
2. To distinguish among image-space and object-space approach for visible-surface determination.
3. To develop and understand the depth-buffer method for visible-surface determination.
4. To develop and explain the Scan-line method for visible-surface determination.
5. To understand the Area-Subdivision method for visible-surface determination.

10.1 VISIBLE-SURFACE DETERMINATION

10.1.1 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 can use 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 with incremental 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 surface can 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 $x_1=x_2$ and $y_1=y_2$ we can say that points are on the same projector.

However, in perspective projection we must perform four divisions: $x_1/z_1=x_2/z_2$ and $y_1/z_1=y_2/z_2$ 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 projector intersects 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.

10.1.2. 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.

10.2 CATEGORIES OF ALGORITHMS

10.2.1 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 between them.
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 $V=(0, 0, Vz)$ and $V.N = Vz C$
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.

10.2.2 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 position along 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.

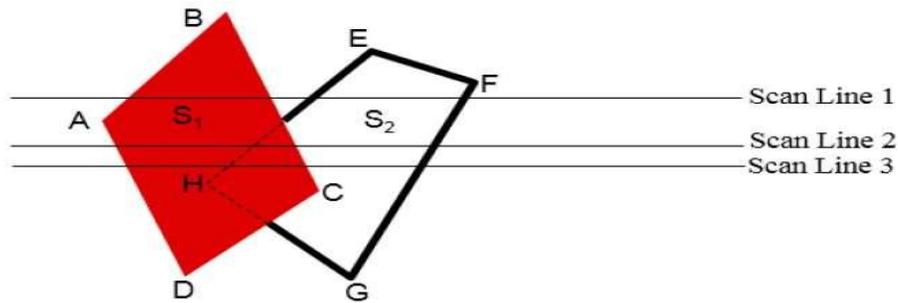
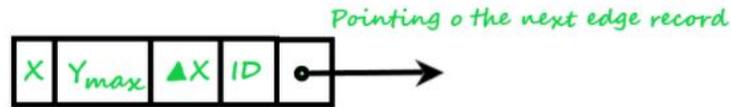


Figure 1: Example 1 Scan Line Algorithm

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 and 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.



$\Delta X = \Delta Y / m = (Y_{max} - Y_{min}) / m = 1/m$,where, $m = \text{slope}$ and
 $\Delta Y = Y_{max} - Y_{min} \cong 1/m$,where Y_{min} and Y_{max} are the two end points Y-coordinate of the edge

ID: Corresponding ID of the edge that has been stored

Figure 2: Edge list table

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)].
8. **Example 2:** Here, two overlapped polygons are given which are intersected by three Scan-lines S_1, S_2, S_3 respectively. So, What happens if the Scan-line algorithm is applied in order to identify the Hidden surface(visible surface).

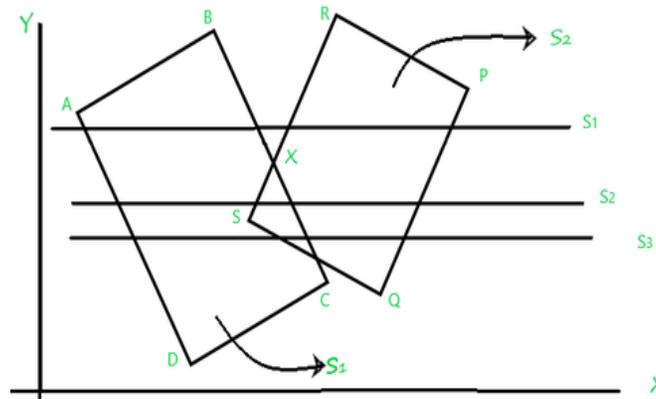


Figure 3: Example 2 Scan Line Algorithm

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, of surface intensity . Next,

If $depth(S_1) > depth(S_2)$, then the Flag of surface $S_1 =$ ” on” and intensity of 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(S_1) and depth(S_2) again so S_3 can take the advantage of the concept of Coherence.

10.2.3 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 calculating the 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 greater that 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.

- 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.

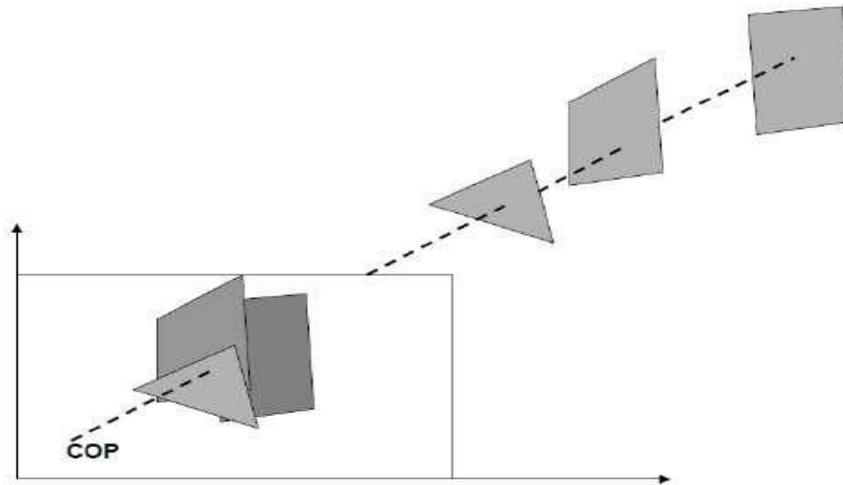


Figure 4: Example 1 Scan Line Algorithm

Algorithm:

- Initialize the Z-buffer and frame buffer so that for all buffer positions $z\text{-buffer}(x,y)=0$ and $\text{Frame-buffer}(x,y)=I_{\text{background}}$
- 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.
- Calculate z -value for each (x,y) position on the polygon
 If $z > z\text{-buffer}(x,y)$ then
 $z\text{-buffer}(x,y)=z,$
 $\text{Frame-buffer}(x,y)= I_{\text{surface}}(x,y).$
- Stop.
 (After processing of all surfaces, the z -buffer contains depth values for the visible surfaces and the frame buffer contain the corresponding intensity values for those surfaces.)

Advantages:

- It is easy to implement
- It can be implemented in hardware to overcome the speed problem.
- Since the algorithm processes objects one at a time, the total number of 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 pixel instead of for the entire polygon.

10.2.4 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 and partly 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.

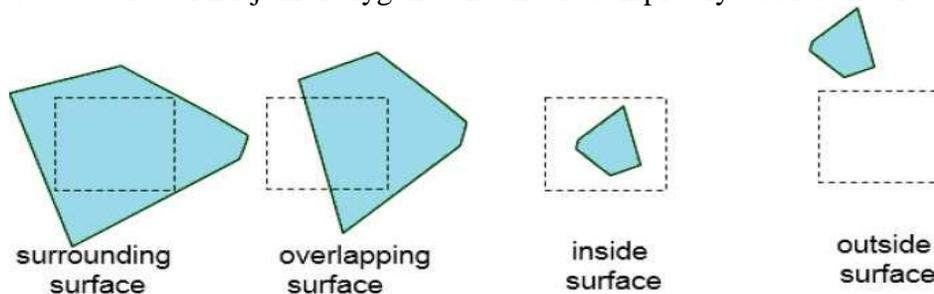


Figure 5: Area Subdivision Algorithm

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.

Fig a) The four intersections of surrounding polygon are all closer to the viewpoint than any of the other intersections. Therefore, the entire area is filled with the colour of the surrounding polygon.

b) That surrounding polygon is not completely in front of the intersecting polygon.

The Warnock's algorithm stops subdivision of area only when the problem is simplified or even area is only a single pixel.

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 to the 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, parallel computers can be used to speed up the process.
2. Extra memory buffer is not required.

10.2.5 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.

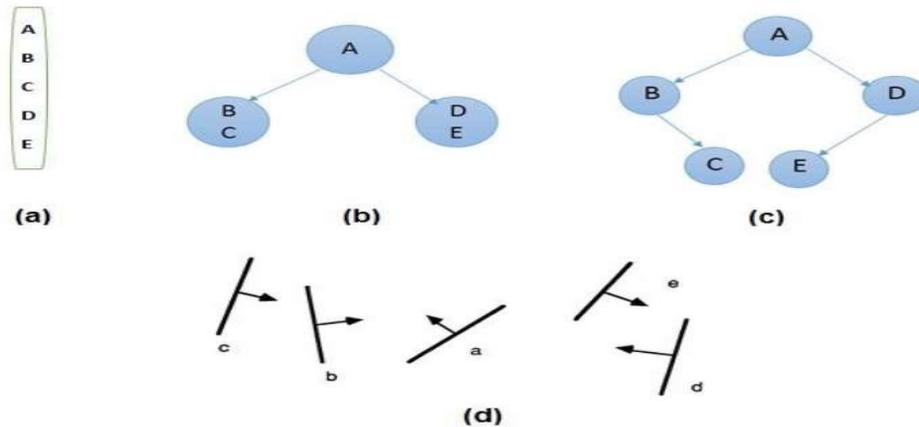


Figure 6: BSP trees

- From the above figure, first take A as a root.
- Make a list of all nodes in figure (a).
- Put all the nodes that are in front of root A to the left side of node A and put all those nodes that are behind the root A to the right side as shown in figure (b).
- Process all the front nodes first and then the nodes at the back.
- As shown in figure (c), we will first process the node B. As there is nothing in front of the node B, we have put NIL. However, we have node C at back of node B, so node C will go to the right side of node B.
- Repeat the same process for the node D.

10.2.6 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 –

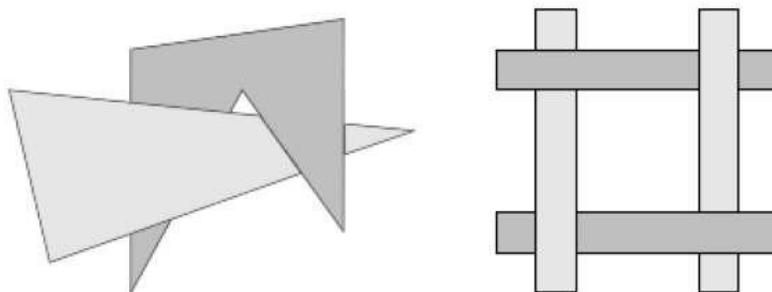


Figure 7: Depth Sorting Method

- 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.
 - Do the x-extents not overlap?
 - Do the y-extents not overlap?
 - Is P entirely on the opposite side of Q’s plane from the viewpoint?
 - Is Q entirely on the same side of P’s plane as the viewpoint?
 - Do the projections of the polygons not overlap?
- If all the tests fail, then we split either P or Q using the plane of the other. The new cut polygons are inserted into the depth order and the process continues. Theoretically, this partitioning could generate $O(n^2)$ individual polygons, but in practice, the number of polygons is much smaller.

10.2.7 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.

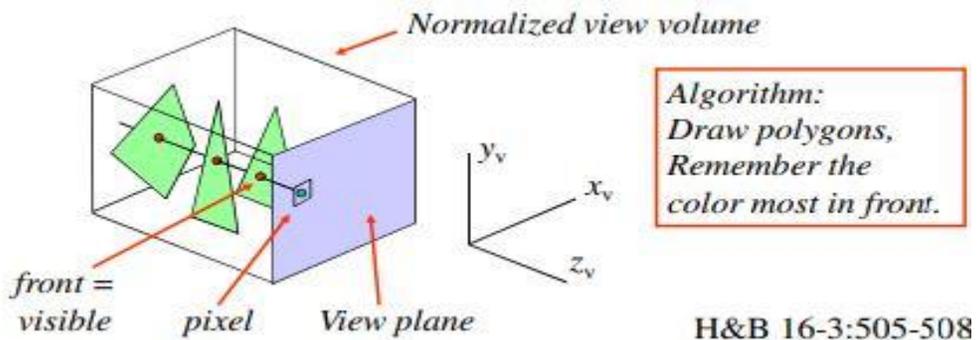


Figure 8: Visible-Surface Ray Tracing

10.3 COMPARISON OF THE METHODS

- Hardware available? Use depth-buffer, possibly in combination with back-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.)

10.4 SUMMARY

When we view a picture containing non-transparent objects and surfaces, then we cannot see those objects from view which are behind from objects closer to eye. We must remove these hidden surfaces to get a realistic screen image. The identification and removal of these surfaces is called Hidden-surface problem. We learn many algorithms to remove hidden surface.

10.5 QUESTIONS

- Q1. Explain Scan-Line Method.
- Q2. Write note on Area-Subdivision Method.
- Q3. Explain Binary Space Partition (BSP) Trees with example
- Q4. What is the concept of Back Face Culling? Explain.
- Q7. Explain Painters algorithm.
- Q8. Differentiate between Ray Casting and Ray Tracing.
- Q9. Describe the methods used for determination of visible surfaces in rendering.
- Q10. Explain techniques for efficient Visible-Surface Algorithms.
- Q11. Explain Image Space Method.
- Q12. Write a Short Note on Object Space Method.

10.6 REFERENCES

- Computer Graphics Hearn, Baker Pearson Education Second
- Steve Marschner, Peter Shirley Fundamentals of Computer Graphics CRC press Fourth Edition 2016.
- A P Godse, Computer Graphics.

PLANE CURVES AND SURFACES

Unit Structure

- 11.0 Objectives
- 11.1 Curve Representation
 - 11.1.1 Nonparametric Curves, Parametric Curves
- 11.2 Parametric Representation of a Circle
- 11.3 Parametric Representation of an Ellipse
- 11.4 Parametric Representation of a Parabola
- 11.5 Parametric Representation of a Hyperbola
- 11.6 Representation of Space Curves
 - 11.6.1 Bezier Curves
 - 11.6.2 B-spline Curves
 - 11.6.3 B-spline Curve Subdivision
 - 11.6.4 Parametric Cubic Curves
 - 11.6.5 Quadric Surfaces
 - 11.6.6 Bezier Surfaces.
- 11.7 Summary
- 11.8 Questions
- 11.9 References
- 11.10 Multiple Choice Questions

11.0 OBJECTIVES

1. To learn curves and surfaces and types of curves.
2. To understand the parametric representation of Curves.
3. To understand Quadratic Surfaces and curves.
4. To understand Bezier curve/surface and their properties.

11.1 INTRODUCTION TO CURVE

In computer graphics, we often need to draw different types of objects onto the screen. Objects are not flat all the time and we need to draw curves many times to draw an object.

11.1.1 Types of Curves:

A curve is an infinitely large set of points. Each point has two neighbours except endpoints. Curves can be broadly classified into three categories – explicit, implicit, and parametric curves.

Implicit Curves:

Implicit curve representations define the set of points on a curve by employing a procedure that can test to see if a point is on the curve. Usually, an implicit curve is defined by an implicit function of the form –

$$f(x, y) = 0$$

It can represent multivalued curves (multiple y values for an x value). A common example is the circle, whose implicit representation is

$$x^2 + y^2 - R^2 = 0$$

Explicit Curves:

A mathematical function $y = f(x)$ can be plotted as a curve. Such a function is the explicit representation of the curve. The explicit representation is not general, since it cannot represent vertical lines and is also single-valued. For each value of x, only a single value of y is normally computed by the function.

Parametric Curves:

Curves having parametric form are called parametric curves. The explicit and implicit curve representations can be used only when the function is known. In practice the parametric curves are used. A two-dimensional parametric curve has the following form –

$$P(t) = (f(t), g(t)) \text{ or } P(t) = (x(t), y(t))$$

The functions f and g become the (x, y) coordinates of any point on the curve, and the points are obtained when the parameter t is varied over a certain interval [a, b], normally [0, 1].

11.2 PARAMETRIC REPRESENTATION OF CURVE CIRCLE

The most common way to write a circle is $x^2 + y^2 = r^2$

Parametrically, this can be written as

$$y(t) = r \cdot \sin(2 \cdot \pi \cdot t)$$

$$x(t) = r \cdot \cos(2 \cdot \pi \cdot t)$$

or if we state the angle in degrees:

$$y(t) = r \cdot \sin(360 \cdot t)$$

$$x(t) = r \cdot \cos(360 \cdot t)$$

when t runs from 0 to 1, the angle argument runs a full circle, and the point (x(t), y(t)) describes a circle.

11.3 PARAMETRIC REPRESENTATION OF ELLIPSE

An ellipse can be defined as the locus of all the points that satisfy the equations:

$$X = a \cos t \quad y = b \sin t$$

Where x, y are the coordinates of any point on the ellipse, a, b are the radius on the x and y axes respectively, t is the parameter which ranges from 0 to 2π radians.

describe the ellipse as $(x/r_x)^2 + (y/r_y)^2 = 1$

Parametrically we describe an ellipse like this:

$$x(t') = r_x \cdot \cos(t')$$

$$y(t') = r_y \cdot \sin(t')$$

$$0 \leq t \leq 2\pi$$

The parametric equations of an ellipse centered at the origin

Recall the construction of a point of an ellipse using two concentric circles of radii equal to lengths of the semi – axes a and b , with the center at the origin as shows the figure, then

$$x = a \cos t$$

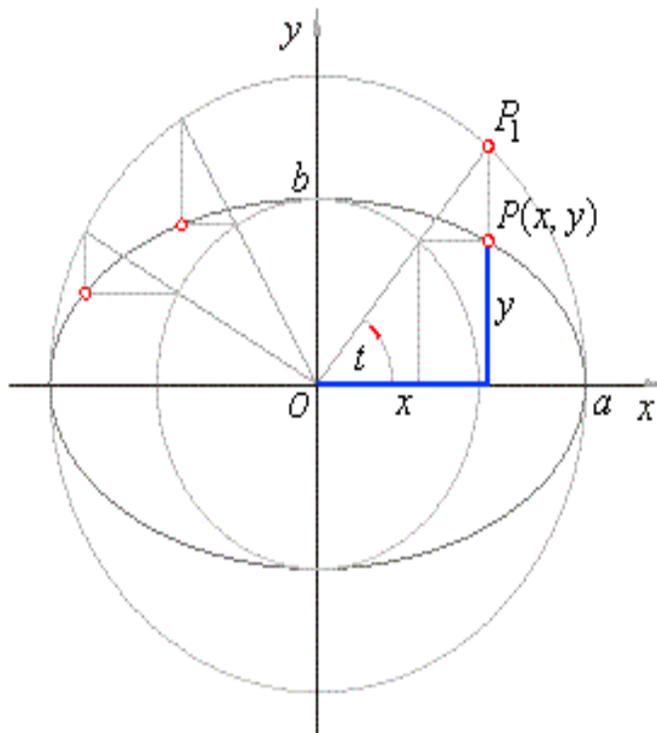
$$y = b \sin t$$

Where, $0 \leq t < 2\pi$

To convert the above parametric equations into Cartesian co-ordinates, divide the first equation by a and the second by b then square and add them.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Thus, obtained is the standard equation of the ellipse.



The parametric equations of a translated ellipse with center at (x_0, y_0)
 The parametric equations of a translated ellipse with center (x_0, y_0) and semi-axes a and b ,

$$x = x_0 + a \cos t$$

$$y = y_0 + b \sin t$$

Where, $0 \leq t < 2\pi$

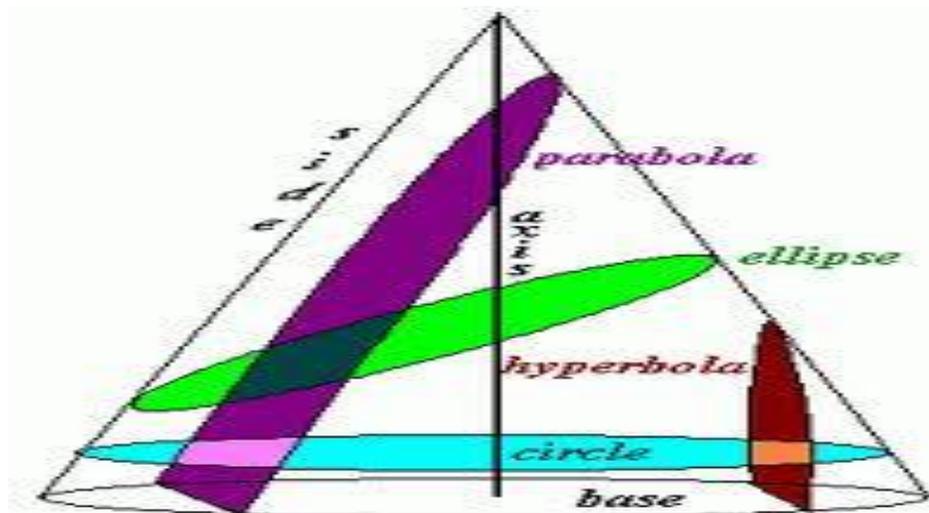
To convert the above parametric equations into Cartesian co-ordinates, we write them as

$$x - x_0 = a \cos t$$

$$y - y_0 = b \sin t$$

And divide the first equation by a and second by b , then square and add both equations, so we get

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$$



11.4 PARAMETRIC REPRESENTATION OF PARABOLA

The parametric equations of a translated ellipse with center at (x_0, y_0)

The parametric equations of a translated ellipse with center (x_0, y_0) and semi-axes a and b ,

$$x = x_0 + a \cos t$$

$$y = y_0 + b \sin t \quad \text{where, } 0 \leq t < 2\pi.$$

To convert the above parametric equations into Cartesian coordinates, we write them as

$$x - x_0 = a \cos t$$

$$y - y_0 = b \sin t$$

and divide the first equation by a and the second by b , then square and add both equations, so we get

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1.$$

Since every parabola is congruent to $x^2 = 4ay$, for some choice of $a > 0$, we can study the geometry of every parabola by confining ourselves to this one. This shows the power of both coordinate geometry and transformations. Imagine a particle moving in the plane along the curve C as shown.

The x - and y - coordinate of the particle can be thought of as functions of a new variable t , and so we can write $x=f(t)$, $y=g(t)$, where f, g are functions of t . t is time.

The equations $x= f(t)$, $y = g(t)$ are called the parametric equations of the point $P(x,y)$ and the variable t is called a parameter.

It is often very useful to take a Cartesian equation $y= F(x)$ and introduce a parameter t so that the x and y – coordinates of any point on the curve can be expressed In terms of this parameter.

11.5 PARAMETRIC REPRESENTATION OF HYPERBOLA

A hyperbola centred at the origin with a left-right opening has the equation:

$$(x^2/a^2) - (y^2/b^2) = 1$$

In this case, we can use the equations $x= a \sec t$, $y= b \tan t$. This is because if you plug these expressions in to the above equation, then the equation reduces to the trigonometric identity $\sec^2 t - \tan^2 t = 1$

Hyperbola centred at the origin with an up-down opening has the equation:

$$(y^2/a^2) - (x^2/b^2) = 1$$

In this case, switch the expressions for x and y , using the equations

$$X= b \tan t, y= a \sec t.$$

If the center of the hyperbola is at (h, k) , then the equation is:

$$((x-h)^2/a^2) - ((y-k)^2/b^2) = 1 \text{ or } ((y-k)^2/a^2) - ((x-h)^2/b^2) = 1$$

In the first case, the parametric equation $x- h = a \sec t$, $y= K + b \tan t$ will work, and in the second case it will be $x-h + b \tan t$, $y=k + a \sec t$.

11.6 REPRESENTATION OF SPACE CURVES

11.6.1 Bezier curve:

Bezier curve is discovered by the French engineer **Pierre Bézier**. These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve. The Bezier curve can be represented mathematically as –

$$\sum_{k=0}^n P_k B_k^n(t)$$

Where p_i is the set of points and $B_i^n(t)$ represents the Bernstein polynomials which are given by –

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i$$

Where n is the polynomial degree, i is the index, and t is the variable.

The simplest Bezier curve is the straight line from the point P_0 to P_1 . A quadratic Bezier curve is determined by three control points. A cubic Bezier curve is determined by four control points.

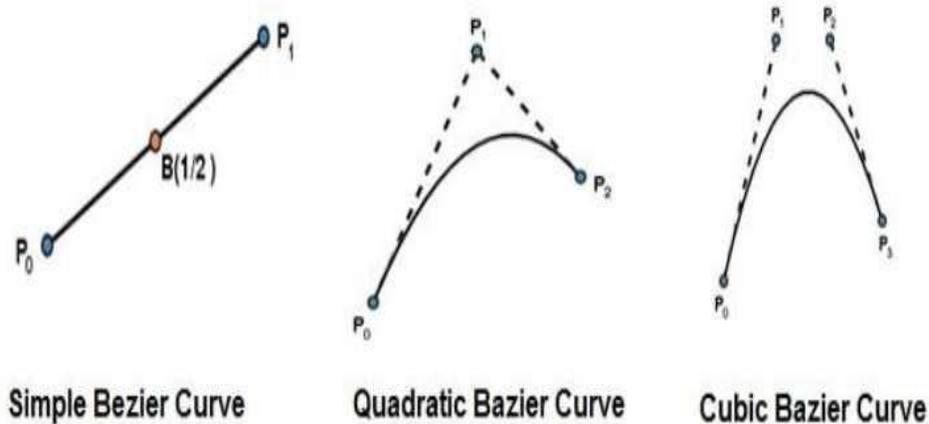


Figure 1: Bezier Curves

11.6.2. Properties of Bezier Curves:

Bezier curves have the following properties:

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point $t=t_0$ into two Bezier segments which join at the point corresponding to the parameter value $t=t_0$.

11.6.3. Applications of Bezier Curves:

Bezier curves have their applications in the following fields-

1. Computer Graphics:

- Bezier curves are widely used in computer graphics to model smooth curves.
- The curve is completely contained in the convex hull of its control points.
- So, the points can be graphically displayed & used to manipulate the curve intuitively.

2. Animation:

- Bezier curves are used to outline movement in animation applications such as Adobe Flash and synfig.
- Users outline the wanted path in bezier curves.
- The application creates the needed frames for the object to move along the path.
- For 3D animation, bezier curves are often used to define 3D paths as well as 2D curves.

3. Fonts:

True type fonts use composite bezier curves composed of quadratic bezier curves.

- Modern imaging systems like postscript, asymptote etc use composite bezier curves composed of cubic bezier curves for drawing curved shapes.

11.6.4. Given a bezier curve with 4 control points- $B_0[1\ 0]$, $B_1[3\ 3]$, $B_2[6\ 3]$, $B_3[8\ 1]$ Determine any 5 points lying on the curve.

Solution: We have-

- The given curve is defined by 4 control points.
- So, the given curve is a cubic bezier curve.

The parametric equation for a cubic bezier curve is-

$$P(t) = B_0(1-t)^3 + B_13t(1-t)^2 + B_23t^2(1-t) + B_3t^3$$

Substituting the control points B_0 , B_1 , B_2 and B_3 , we get-

$$P(t) = [1\ 0](1-t)^3 + [3\ 3]3t(1-t)^2 + [6\ 3]3t^2(1-t) + [8\ 1]t^3 \dots\dots(1)$$

Now,

To get 5 points lying on the curve, assume any 5 values of t lying in the range $0 \leq t \leq 1$.

Let 5 values of t are 0, 0.2, 0.5, 0.7, 1

For $t = 0$:

Substituting $t=0$ in (1), we get-

$$P(0) = [1\ 0](1-0)^3 + [3\ 3]3(0)(1-t)^2 + [6\ 3]3(0)^2(1-0) + [8\ 1](0)^3$$

$$P(0) = [1 \ 0] + 0 + 0 + 0$$

$$P(0) = [1 \ 0]$$

For $t = 0.2$:

Substituting $t=0.2$ in (1), we get-

$$P(0.2) = [1 \ 0](1-0.2)^3 + [3 \ 3]3(0.2)(1-0.2)^2 + [6 \ 3]3(0.2)^2(1-0.2) + [8 \ 1](0.2)^3$$

$$P(0.2) = [1 \ 0](0.8)^3 + [3 \ 3]3(0.2)(0.8)^2 + [6 \ 3]3(0.2)^2(0.8) + [8 \ 1](0.2)^3$$

$$P(0.2) = [1 \ 0] \times 0.512 + [3 \ 3] \times 3 \times 0.2 \times 0.64 + [6 \ 3] \times 3 \times 0.04 \times 0.8 + [8 \ 1] \times 0.008$$

$$P(0.2) = [1 \ 0] \times 0.512 + [3 \ 3] \times 0.384 + [6 \ 3] \times 0.096 + [8 \ 1] \times 0.008$$

$$P(0.2) = [0.512 \ 0] + [1.152 \ 1.152] + [0.576 \ 0.288] + [0.064 \ 0.008]$$

$$P(0.2) = [2.304 \ 1.448]$$

For $t = 0.5$:

Substituting $t=0.5$ in (1), we get-

$$P(0.5) = [1 \ 0](1-0.5)^3 + [3 \ 3]3(0.5)(1-0.5)^2 + [6 \ 3]3(0.5)^2(1-0.5) + [8 \ 1](0.5)^3$$

$$P(0.5) = [1 \ 0](0.5)^3 + [3 \ 3]3(0.5)(0.5)^2 + [6 \ 3]3(0.5)^2(0.5) + [8 \ 1](0.5)^3$$

$$P(0.5) = [1 \ 0] \times 0.125 + [3 \ 3] \times 3 \times 0.5 \times 0.25 + [6 \ 3] \times 3 \times 0.25 \times 0.5 + [8 \ 1] \times 0.125$$

$$P(0.5) = [1 \ 0] \times 0.125 + [3 \ 3] \times 0.375 + [6 \ 3] \times 0.375 + [8 \ 1] \times 0.125$$

$$P(0.5) = [0.125 \ 0] + [1.125 \ 1.125] + [2.25 \ 1.125] + [1 \ 0.125]$$

$$P(0.5) = [4.5 \ 2.375]$$

For $t = 0.7$:

Substituting $t=0.7$ in (1), we get-

$$P(t) = [1 \ 0](1-t)^3 + [3 \ 3]3t(1-t)^2 + [6 \ 3]3t^2(1-t) + [8 \ 1]t^3$$

$$P(0.7) = [1 \ 0](1-0.7)^3 + [3 \ 3]3(0.7)(1-0.7)^2 + [6 \ 3]3(0.7)^2(1-0.7) + [8 \ 1](0.7)^3$$

$$P(0.7) = [1 \ 0](0.3)^3 + [3 \ 3]3(0.7)(0.3)^2 + [6 \ 3]3(0.7)^2(0.3) + [8 \ 1](0.7)^3$$

$$P(0.7) = [1 \ 0] \times 0.027 + [3 \ 3] \times 3 \times 0.7 \times 0.09 + [6 \ 3] \times 3 \times 0.49 \times 0.3 + [8 \ 1] \times 0.343$$

$$P(0.7) = [1 \ 0] \times 0.027 + [3 \ 3] \times 0.189 + [6 \ 3] \times 0.441 + [8 \ 1] \times 0.343$$

$$P(0.7) = [0.027 \ 0] + [0.567 \ 0.567] + [2.646 \ 1.323] + [2.744 \ 0.343]$$

$$P(0.7) = [5.984 \ 2.233]$$

For $t = 1$:

Substituting $t=1$ in (1), we get-

$$P(1) = [1 \ 0](1-1)^3 + [3 \ 3]3(1)(1-1)^2 + [6 \ 3]3(1)^2(1-1) + [8 \ 1](1)^3$$

$$P(1) = [1 \ 0] \times 0 + [3 \ 3] \times 3 \times 1 \times 0 + [6 \ 3] \times 3 \times 1 \times 0 + [8 \ 1] \times 1$$

$$P(1) = 0 + 0 + 0 + [8 \ 1]$$

$$P(1) = [8 \ 1]$$

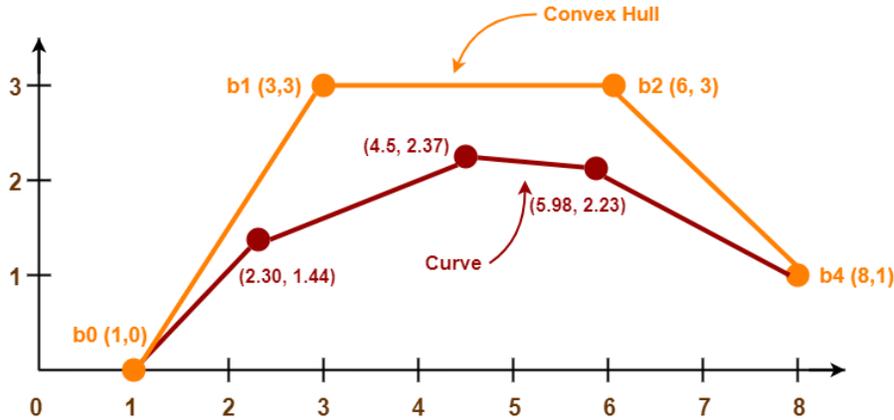


Figure 2: Example of Bezier Curve

11.6.2 B-Spline Curves:

The Bezier-curve produced by the Bernstein basis function has limited flexibility.

- First, the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
- The second limiting characteristic is that the value of the blending function is nonzero for all parameter values over the entire curve.

The B-spline basis contains the Bernstein basis as the special case. The B-spline basis is non-global.

A B-spline curve is defined as a linear combination of control points P_i and B-spline basis function $N_{i,k}(t)$ given by

$$C(t) = \sum_{i=0}^n P_i N_{i,k}(t), \quad n \geq k-1, \quad t \in [tk-1, tn+1]$$

Where,

- $\{P_i: i=0, 1, 2, \dots, n\}$ are the control points
- k is the order of the polynomial segments of the B-spline curve. Order k means that the curve is made up of piecewise polynomial segments of degree $k-1$,
- the $N_{i,k}(t)$ are the “normalized B-spline blending functions”. They are described by the order k and by a non-decreasing sequence of real numbers normally called the “knot sequence”.

$$t_i: i=0, \dots, n+K$$

The $N_{i,k}$ functions are described as follows –

$$N_{i,1}(t) = \begin{cases} 1, & \text{if } t \in [t_i, t_{i+1}) \\ 0, & \text{Otherwise} \end{cases}$$

and if $k > 1$,

$$N_{i,k}(t) = (t - t_i / t_{i+k-1}) N_{i,k-1}(t) + (t_{i+k} - t / t_{i+k} - t_{i+1}) N_{i+1,k-1}(t)$$

and

$t \in [t_{k-1}, t_{n+1})$

Properties of B-spline Curve:

B-spline curves have the following properties:

- The sum of the B-spline basis functions for any parameter value is 1.
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for $k=1$.
- The maximum order of the curve is equal to the number of vertices of defining polygon.
- The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.
- The curve exhibits the variation diminishing property.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon

11.6.3 B-Spline Curve Subdivision :

Subdivision “Subdivision defines a smooth curve or surface as the limit of a sequence of successive refinements”

The sub division of B- Spline curve used De Boor’s Algorithm. It provides a fast and numerically stable way for finding a point on a B-spline curve given unit domain. It is the algorithm for evaluating spline curves in B-spline forms.

Selecting Control Points:

Suppose we want to subdivide a B-spline curve at u into two B-spline curves, one on $[0,u]$ and the other on $[u,1]$. The first thing to do is to apply de Boor's algorithm at u . Note that if u is in $[u_k, u_{k+1})$, then at most $p+1$ control points will be used in the computation, namely: $P_k, P_{k-1}, \dots, P_{k-p}$. To find the control points of the B-spline curve on $[0,u]$, we start with P_0 and follow the control polyline. When we encounter a control point or a point computed in the process of de Boor's algorithm, we select that point and make a turn. This process continues until we reach $C(u)$ which is also selected as a control point. These points, in the visiting order, form the control points that define the B-spline curve on $[0,u]$. For the B-spline curve defined on $[u,1]$, we start with the point $C(u)$ and apply the same process. We select each encountered point and make a turn until P_n is reached.

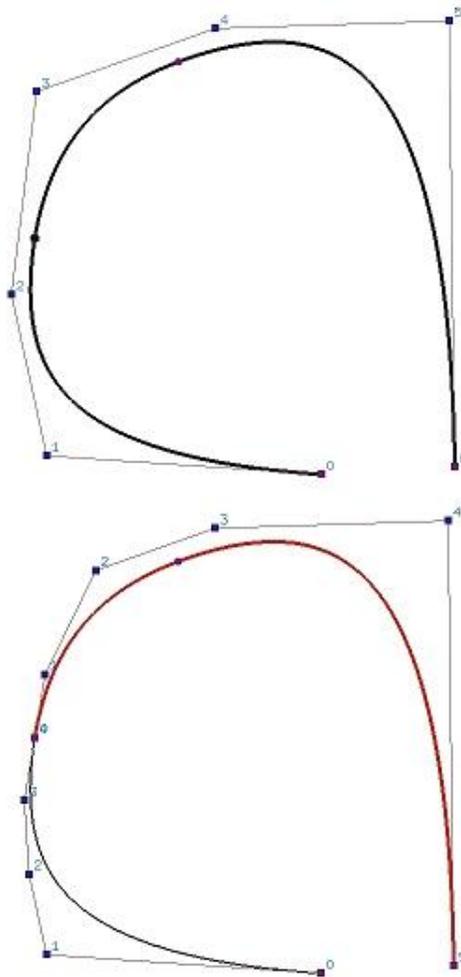


Figure 3: Plotting of Control Points

11.6.4 Parametric Cubic curves:

Cubic curves are commonly used in graphics because curves of lower order commonly have too little flexibility, while curves of higher order are usually considered unnecessarily complex and make it easy to introduce undesired wiggles. A parametric cubic curve in 3D is defined by: usually, we consider $t = [0 \dots 1]$.

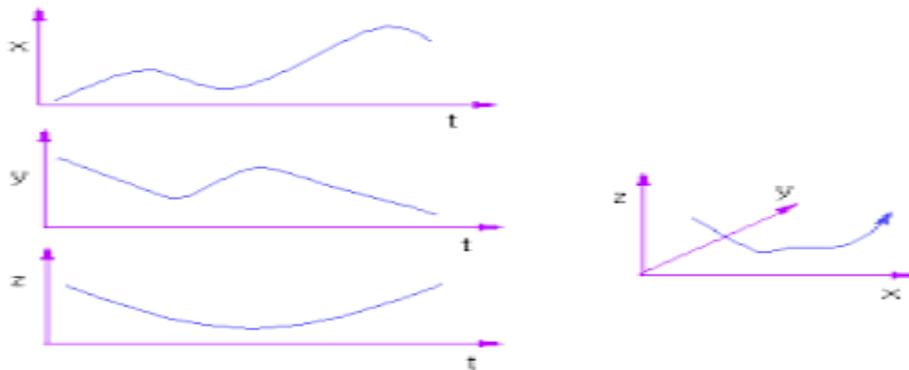


Figure 4: Cubic Curves

11.6.5 Quadratic Surfaces:

A second order algebraic surface given by the general equation:

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0.$$

Examples of quadratic surfaces include the cone, cylinder, ellipsoid, elliptic cone, elliptic cylinder, hyperbolic cylinder, hyperbolic paraboloid, paraboloid, sphere and spheroid.

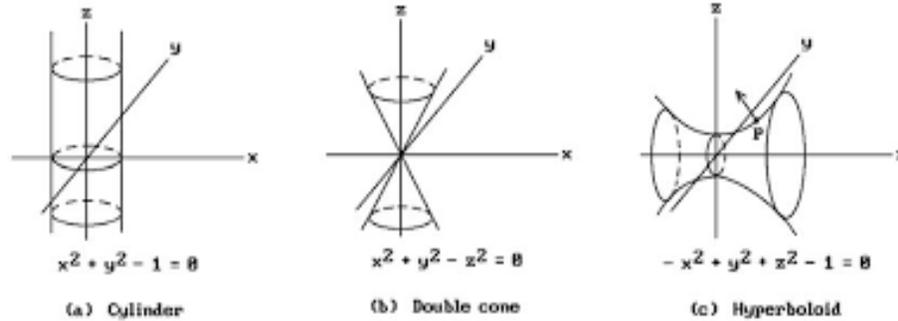


Figure 5: Quadratic Surfaces

11.6.6 Bezier surface:

Bezier surfaces are a species of mathematical spline used in computer graphics, computer-aided design, and finite element modeling.

- Bezier surfaces are an extension of the idea of Bezier curves and share many of their properties.
- Bezier surfaces are defined by a set of control points.
- Properties are-
 - the surface does not in general pass through the control points (except for the corner points)
 - the surface is contained within the convex hull of the control points.

A two-dimensional Bézier surface can be defined as a parametric surface where the position of a point \mathbf{p} as a function of the parametric coordinates u, v is given by:

$$\mathbf{P}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) \mathbf{k}_{i,j}$$

evaluated over the unit square, where

$$B_i^n(u) = \binom{n}{i} u^i (1 - u)^{n-i}$$

is a Bernstein polynomial, and

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \text{ is the binomial coefficient.}$$

Some properties of Bézier surfaces:

- A Bézier surface will transform in the same way as its control points under all linear transformations and translations.

- All $u = \text{constant}$ and $v = \text{constant}$ lines in the (u, v) space, and all four edges of the deformed (u, v) unit square are Bézier curves.
- A Bézier surface will lie completely within the convex hull of its control points, and therefore also completely within the bounding box of its control points in any given Cartesian coordinate system.
- The points in the patch corresponding to the corners of the deformed unit square coincide with four of the control points.
- However, a Bézier surface does not generally pass through its other control points.

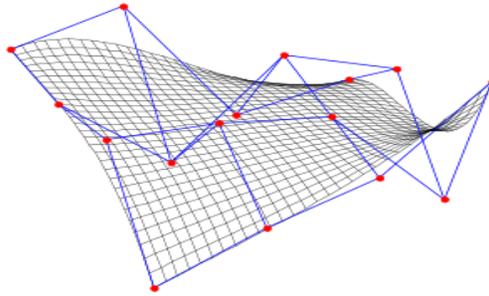


Figure 6: Bézier surface

Sample Bézier surface; red - control points, blue - control grid, black - surface approximation

11.7 SUMMARY

When we view a picture containing non-transparent objects and surfaces, then we cannot see those objects from view which are behind from objects closer to eye. We must remove these hidden surfaces to get a realistic screen image. The identification and removal of these surfaces is called Hidden-surface problem. We learn many algorithms to remove hidden surface and along with that different types of curves and surface with its types.

11.8 QUESTIONS

- Q1. Explain Parametric Representation of a Circle.
- Q2. Explain Parametric Representation of an Ellipse.
- Q3. Explain Parametric Representation of a Parabola and Hyperbola
- Q4. Explain Bezier Curves and surfaces.
- Q5. Define Curve. What are its types? Explain in brief.
- Q6. What are the properties of Bezier Curves? Explain.
- Q7. Explain Bezier Curves properties.
- Q9. Write a short note on Quadratic Surfaces.
- Q10. Explain Bezier Surfaces with example.

11.9 REFERENCES

1. Computer Graphics Hearn, Baker Pearson Education Second
2. Steve Marschner, Peter Shirley Fundamentals of Computer Graphics CRC press Fourth Edition 2016.
3. A P Godse, Computer Graphics.

11.10 MULTIPLE CHOICE QUESTIONS

1. In _____ method, visibility is decided point by point at each pixel position on the projection plane.
 - a. **Image-Space**
 - b. Object-Space
 - c. Polygon Clipping
 - d. Coherence
2. Z-buffer algorithm is used for _____.
 - a. Frame buffer removal
 - b. **Hidden line removal**
 - c. Rendering
 - d. Animation
3. _____ is fast and effective as an initial screening to eliminate many polygons from further visibility tests.
 - a. **Back-face detection**
 - b. Painter's algorithm
 - c. Z-buffer algorithm
 - d. Area Sub-division algorithm
4. The _____ is particularly useful when the view reference point changes, but the objects in a scene are at fixed positions.
 - a. **BSP Tree**
 - b. Back-face
 - c. Painter's algorithm
 - d. Z-buffer algorithm
5. _____ method successively divides the total viewing area into smaller and smaller rectangles until each small area is the projection of part of a single visible surface or no surface at all.
 - a. **Area-subdivision method**
 - b. Back-face
 - c. Painter's algorithm
 - d. Z-buffer algorithm
6. Which of the following method is not a image space method?
 - a. **Back-face Detection**
 - b. Z Buffer Algorithm

- c. Scan line algorithm
 - d. Ray Tracing
7. Most visible-surface detection algorithms use sorting and _____ methods to improve performance.
- a. **Coherence**
 - b. Object-Space
 - c. Polygon Clipping
 - d. Dept information
8. Visible Surface Detection Method is also called as _____.
- a. **Hidden-Surface elimination method**
 - b. Hidden- Surface Identification Method
 - c. Hidden-Surface Detection Method
 - d. VSD
9. Z-buffer algorithm uses 2 buffer, depth buffer for storing depth values and _____ buffer stores intensity values at each pixel position.
- a. **Refresh**
 - b. Intensity
 - c. Color
 - d. Depth
10. _____ curves must precisely have n control points.
- a. **Bezier**
 - b. B-Spline
 - c. Curved
 - d. Parametric

UNIT V

12

COMPUTER ANIMATION

Unit Structure

- 12.0 Objectives
- 12.1 Introduction
- 12.2 Definition
- 12.3 Principles of Animation
 - 12.3.1 Squash and Stretch
 - 12.3.2 Anticipation
 - 12.3.3 Staging
 - 12.3.4 Straight Ahead Action and Pose to Pose
 - 12.3.5 Follow Through and Overlapping Action
 - 12.3.6 Ease In and Ease Out
 - 12.3.7 Arc
 - 12.3.8 Secondary Action
 - 12.3.9 Timing
 - 12.3.10 Exaggeration
 - 12.3.11 Solid drawing
 - 12.3.12 Appeal
- 12.4 Key Frame Animation
- 12.5 Deformation
- 12.6 Character Animation
- 12.7 Physics Based Animation
- 12.8 Procedural Techniques
- 12.9 Groups Of objects
- 12.10 Summary
- 12.11 References
- 12.12 Bibliography
- 12.13 Exercise

12.0 OBJECTIVES

After this chapter, student will be able to:

- Understand the computer animation and examples of animation.
- Understand the principles of animation.

- Study the types of animation like key frame animation, character animation, Physics based animation.
- Understand the process of deformation and its requirement.
- Understand the concept behind object groups.

12.1 INTRODUCTION

- Computer animation is growing industry in the area of Web, TV, games, multimedia, films, digital art, architectural visualization etc.
- The goal of computer animation is to process the desired motion effect by mixing perception, natural information and imagination.
- It helps us to understand physical law by adding motion control to data.
- Many people can use computer animation in profession like pilots, architect, surgeons, automobile company, gaming industry etc.

12.2 DEFINITION

- Computer animation may be defined as a technique in which the illusion of a movement is created by displaying on a screen or recording on device individual states of a dynamic scene.
- Any computer animation sequence may be defined as a set of objects.
- It is characterized by state variables evolving over time or it is a time sequence of changes in a scene.
- For example, a human character is normally characterized using its joint angles as state variables.
- Animation is production of consecutive images which when displayed together convey a feeling of motion.
- In addition to changing object position with translations or rotations, a computer generated animation could display time variations in object size, color, texture etc.
- Computer animations can also be generated by changing camera parameters such as position, orientation and focal length.

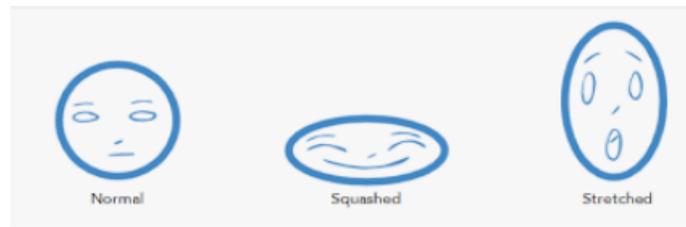


12.3 PRINCIPLES OF ANIMATION

Principles of animation can be categorized in following types:

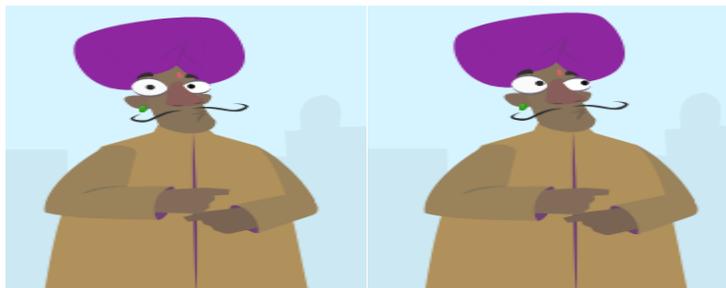
12.3.1 SQUASH AND STRETCH:

- The most important principle is "squash and stretch".
- The purpose of this is to give a sense of weight and flexibility to drawn objects.
- It can be applied to simple objects, example like a bouncing ball, construction like the musculature of a human face etc.
- A figure stretched or squashed to an exaggerated degree can have a comical effect.
- In realistic animation, the most important aspect of this principle is the fact that an object's volume does not change when squashed or stretched.
- If the length of a ball is stretched vertically, its width (in three dimensions and also its depth) needs to contract correspondingly horizontally.



12.3.2 ANTICIPATION :

- Anticipation is used to prepare the audience for an action.
- It makes the action appear more realistic.
- A dancer jumping off the floor has to bend the knees first, a golfer making a swing has to swing the club back first are example of it.



12.3.3 STAGING :

- Its purpose is to direct the audience's attention.
- This makes it clear that what is of greatest importance in a scene.
- Johnston and Thomas defined it as "the presentation of any idea so that it is completely and unmistakably clear", whether that idea is an action, a personality, an expression or a mood.
- This can be done by various means such as the placement of a character in the frame, the use of light and shadow or the angle and position of the camera.
- The essence of this principle is keeping focus on what is relevant.



12.3.4 STRAIGHT AHEAD ACTION AND POSE TO POSE :

These are two different approaches to the actual drawing process.

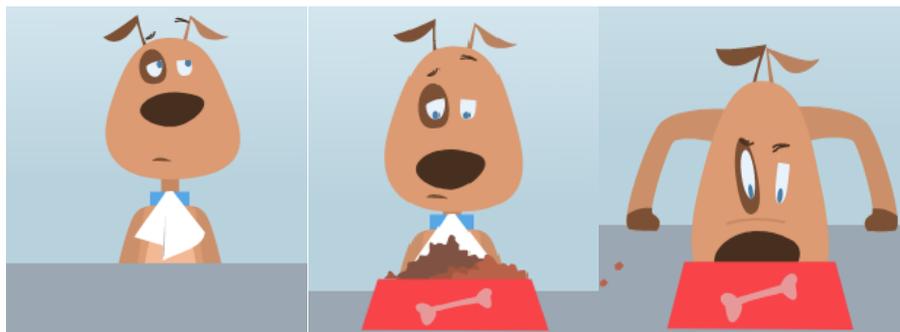
- "Straight ahead action" scenes are animated frame by frame from beginning to end.
- Straight ahead action creates a more fluid, dynamic illusion of movement.
- It is better for producing realistic action sequences.
- It is hard to maintain proportions and to create exact, convincing poses along the way.
- "Pose to pose" involves starting with drawing a few key frames and then filling in the intervals later.
- "Pose to pose" works better for dramatic or emotional scenes.
- Here composition and relation to the surroundings are of greater importance.



12.3.5 FOLLOW THROUGH AND OVERLAPPING ACTION :

Follow through and overlapping action is a general heading for two closely related techniques.

- It helps to render movement more realistically.
- "Follow through" means that loosely tied parts of a body should continue moving after the character has stopped.
- Also the parts should keep moving beyond the point where the character stopped only to be subsequently "pulled back towards the center of mass or exhibiting various degrees of oscillation damping.
- "Overlapping action" is the tendency for parts of the body to move at different rates example an arm will move on different timing of the head and so on.
- A third related technique is "drag" where a character starts to move and parts of him take a few frames to catch up.
- These parts can be inanimate objects like clothing or the antenna on a car or parts of the body, such as arms or hair.



12.3.6 EASE IN AND EASE OUT:

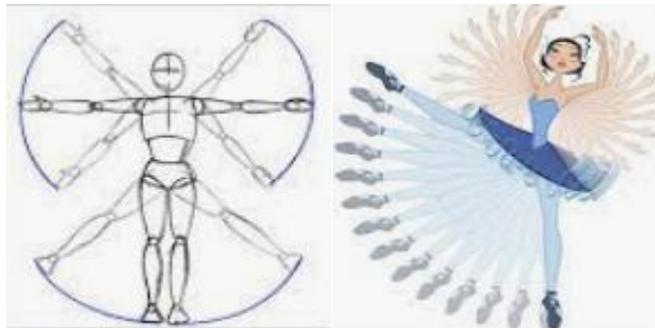
- The movement of the human body and most other objects needs time to accelerate and slow down.
- For this reason, animation looks more realistic.
- It has more drawings near the beginning and end of an action, emphasizing the extreme poses and fewer in the middle.

- This principle goes for characters moving between two extreme poses for example sitting down and standing up but also for inanimate moving objects like the bouncing ball.



12.3.7 ARC:

- Most natural action tends to follow an arched trajectory and animation should adhere to this principle by following implied "arcs" for greater realism.
- This technique can be applied to a moving limb by rotating a joint or a thrown object moving along a parabolic trajectory.
- As an object's speed or momentum increases, arcs tend to flatten out in moving ahead and broaden in turns.
- An object in motion that moves out of its natural arc for no apparent reason will appear erratic rather than fluid.
- For example, when animating a pointing finger, the animator should be certain that in all drawings in between the two extreme poses the fingertip follows a logical arc from one extreme to the next.



12.3.8 SECONDARY ACTION:

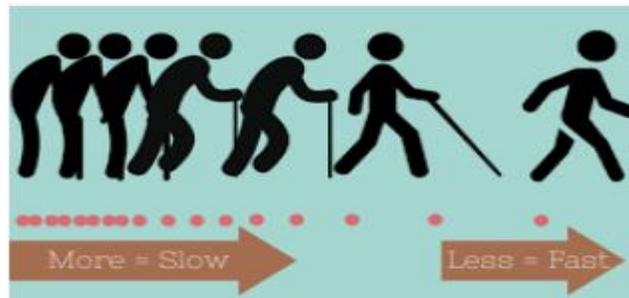
- Adding secondary actions to the main action gives a scene more life.
- It can help to support the main action.
- A person walking can simultaneously swing their arms or keep them in their pockets, speak or whistle or express emotions through facial expressions.
- The important thing about secondary actions is that they emphasize rather than take attention away from the main action.

- For example during a dramatic movement, facial expressions will often go unnoticed.
- In these cases it is better to include them at the beginning and the end of the movement rather than during.



12.3.9 TIMING:

- Timing refers to the number of drawings or frames for a given action.
- It translates to the speed of the action on film.
- Correct timing makes objects appear to obey the laws of physics.
- Example an object's weight determines how it reacts to an impetus like a push.
- Timing is critical for establishing a character's mood, emotion and reaction.
- It can also be a device to communicate aspects of a character's personality.



12.3.10 EXAGGERATION:

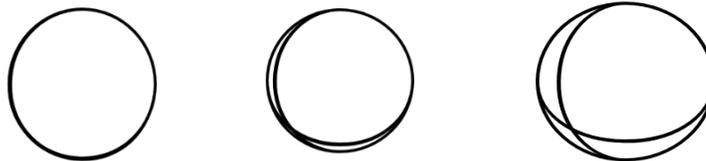
- Exaggeration is an effect especially useful for animation.
- It enhances animated motions that strive for a perfect imitation of reality can look static and dull.
- The level of exaggeration depends on whether one seeks realism or a particular style like a caricature or the style of a specific artist.

- The classical definition of exaggeration employed by Disney was to remain true to reality just presenting it in a wilder and in more extreme form.
- If a scene contains several elements, there should be a balance in how those elements are exaggerated in relation to each other.
- This is to avoid confusing or overawing the viewer.



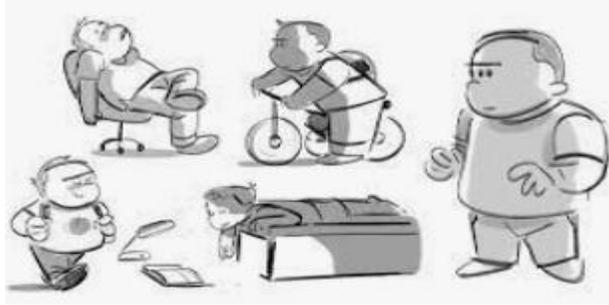
12.3.11 SOLID DRAWING:

- The principle of solid drawing means taking into account forms in three-dimensional space or giving them volume and weight.
- The animator needs to be a skilled artist.
- He has to understand the basics of three-dimensional shapes, anatomy, weight, balance, light and shadow etc.



12.3.12 APPEAL:

- Appeal in a cartoon character corresponds to what would be called charisma in an actor.
- A character that is appealing is not necessarily sympathetic.
- Villains or monsters can also be appealing.
- The important thing is that the viewer feels the character is real and interesting.
- There is way for making a character connect better with the audience.
- Example a complicated or hard to read face will lack appeal.
- It may more accurately be described as 'captivation' in the composition of the pose or the character design.



12.4 KEY-FRAME ANIMATION

- A key frame animation consists of an automatic generation of the intermediate frames based on a set of key frames that is supplied by an animator.
- A **key frame** in animation and filmmaking is a drawing that defines the starting and ending points of any smooth transition.
- In order to produce a key-frame animation the animator creates the behavior of a model manually by using an intuitive the “put that there” methodology.
- The animator has full and direct control over the positions, shapes and motions of models during the animation.
- There are two fundamental approaches to a key-frame:

1. In-betweens are obtained by shape interpolation.

This method makes it possible to transform a geometrical form into another during an animation.

There is a serious problem for an image based key frame as the motion can be distorted due to the interpolation.

The problem becomes complicated when the two drawings do not have the same number of vertices.

In such case it is initially necessary to carry out a preprocessing step to equalize the number of vertices of the two drawings.

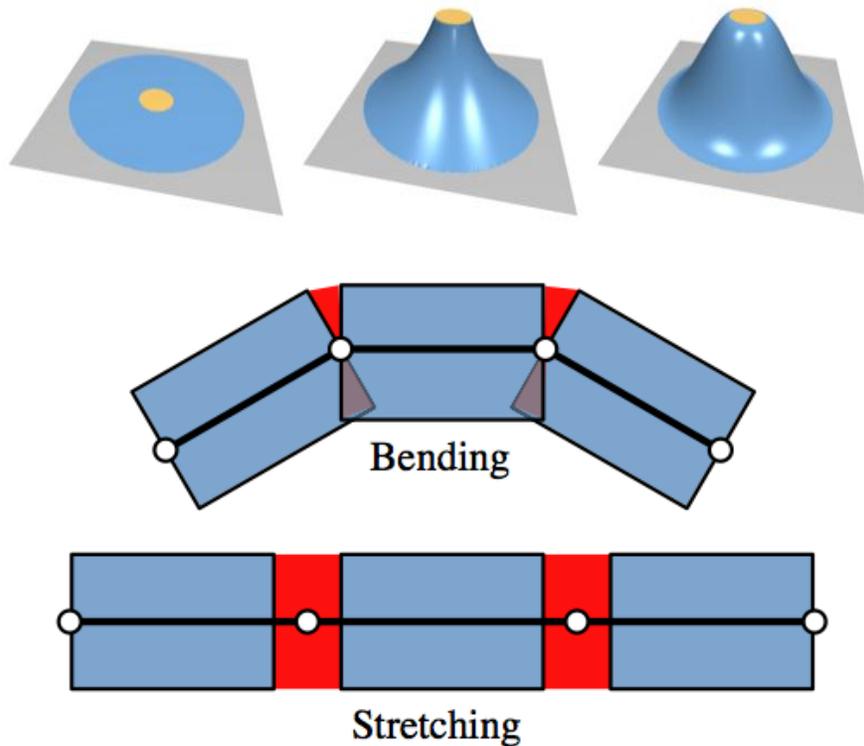
2. A way of producing better images is to interpolate parameters of the model instead of the object itself.

The parameters are normally spatial parameters, physical parameters and visualization parameters that decide the model's behavior.

12.5 DEFORMATION

- It's an important aspect of geometry processing used in CAD and movie industry.
- In this shape manipulation is driven by a variation optimization.

- It guarantees high quality deformations by minimizing physically inspired energies subject to user controlled constraints.
- Physically-plausible deformations are achieved by a constrained minimization of stretching (center) or bending (right) energies.
- In deformations on a very large model user deforms the shape using reduced model.



12.6 CHARACTER ANIMATION:

- This type of animation defines specialized area of animation in which animated character are brought to life.
- Character animators are often known as “actors with pencil”.
- Character animator captures the performance data with varies input technology using camera, keyboard, microphone, mouse or touch devices etc.
- They capture facial expression.
- This make animation looks real.
- Character animator tries to put life in their character work for creating the illusion of emotion, personality and thoughts.
- Example of this are animated feature films, gaming industry,

12.7 PHYSICS BASED ANIMATION

- Physics based animation is study of simulation of physically plausible behaviors at collaborative rate.
- This animation is constraint by strict time as it is imposed by interactive animation.
- It is inspired by need to include physically motivated behaviors in complex movie, video games etc.
- In these animation an animator provides physical data and the motion is obtained by solving the dynamics equations.
- It focuses on numerical stability and visual appeal over the physical accuracy.
- Here motion is globally controlled.
- We can distinguish methods based on parameter like adjustment and constraint-based methods.
- Here the animator states are in terms of constraints and the properties the model is supposed to have without needing to adjust parameters.

12.8 PROCEDURAL TECHNIQUES

- A procedural technique helps a type of computer animation which is used to automatically generate animation in real time.
- It allow for a more diverse series of actions than could also be created using predefined animations.
- The procedural technique is used to simulate particle system like smoke, fire, water, etc., cloth and clothing, rigid-body dynamics, hair and fur dynamics and character animation.
- In computer and video games it is often used for simple things like turning a character's head when a player looks around.
- The procedural techniques correspond to the creation of a motion by a procedure by describing the motion.
- Rules are established for a system and an initial state is defined for objects in the system.
- The procedural technique can be very useful for generating much life-like motion from relatively little input.
- Such an animation should be used when the motion can be described by an algorithm or a formula.
- The generation of a motion using a procedure is not really a method but more a framework.

- There are so many possibilities of expressing a motion using a procedure or a law.
- The number of all possible algorithms is unlimited.
- Procedural approaches can suffer from a lack of explicit control over the look of individual frames.

12.9 GROUPS OF OBJECT

- We should know how to group a set of objects together to apply a single animation.
- To animate multiple objects you have to group them first.
- First right click the selected object and then click on Group > Group.
- Next on animation tab, click on effect that you want to apply to group.
- Apply your animation effect only once.
- Next once finished, when you play back the animation the group will behave like one object.
- If you want to ungroup the objects, click on ungroup option.
- There is also another way of grouping the objects called as drawing a marquee.
- Example in PowerPoint you can animate the group of objects such as clip arts, shapes and pictures.
- This helps to draw audience attention to specific content or to make it easier to read.

12.10 SUMMARY

- Computer animation is art of creating moving images or objects with the help of computer.
- Computer animation is the field area of computer graphics.
- Animator is able to tell story layout and communicate emotions and ideas by applying the principles of animation.
- The simplest form of animating the objects can be done by key frame animation.
- Animation consist of many types such as character animation, key frame animation, physics based animation and so on.
- Procedural techniques are used to generate procedural animation.
- Techniques like deformation and grouping of objects are used in animation to make a study easy.

12.11 EXERCISE

1. Write a short not on key frame animation.
2. Define computer animation. State and explain principles of animation.
3. Explain the concept of deformation.
4. State and explain procedural techniques in animation.
5. Write a short note on character animation and physics based animation.
6. Explain the concept of how to group the objects.

12.12 REFERENCES

1. *Computer Graphics*, R. K. Maurya, John Wiley.

12.13 BIBLIOGRAPHY

1. D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Edition, 1997.
2. J. Hughes, A. Van Dam, M. MCGuire, D. Sklar, J. Foley, and K. Akeley, *Computer Graphics, Principles and Practice*, 3rd Edition, Pearson, 2014.
3. P. Shirley and S. Marschner, "Fundamentals of Computer Graphics: Ray Tracing," *CRC Press*. p. 752, 2009.

UNIT V

13

IMAGE MANIPULATION AND STORAGE

Unit Structure

- 13.0 Objectives
- 13.1 Introduction
- 13.2 What is an Image?
 - 13.2.1 Image Types
- 13.3 Digital Image File Formats
- 13.4 Image Compression
 - 13.4.1 Image Compression Standard-JPEG
- 13.5 Image processing
 - 13.5.1 Digital Image Enhancement
 - 13.5.2 Contrast Stretching
 - 13.5.3 Histogram
 - 13.5.3.1 Histogram Equalization
 - 13.5.4 Filtering
 - 13.5.4.1 Smoothing
 - 13.5.4.1.1 Low Pass Averaging Filter
 - 13.5.4.1.2 Low Pass Median Filter
- 13.6 Summary
- 13.7 References
- 13.8 Bibliography
- 13.9 Exercise

13.0 OBJECTIVES

After this chapter, student will be able to:

- Understand two-dimensional image and image processing system.
- Understand different image file format available.
- Study the image enhancement techniques to perform subjective improvisation of images and filtering techniques in spatial domain.
- Understand image compression standard for optimization of storage.
- Understand histogram and how to find histogram of image by equalization method.

13.1 INTRODUCTION

- Human being depends on eye to capture the information that is around them.
- Among the five senses human beings have, we depend on sight the most.
- There is some example which shows that not every animal depends on visual system as human beings do.
- Example of such animals is bats which use high frequency sound waves or cats that have excellent sense of smell in spite of poor vision and many more.

13.2 WHAT IS AN IMAGE?

- An image can be defined as a 2-dimensional representation of 3-dimensional world.
- The world around us is 3-dimensional
- Images obtained through a camera are 2-dimensional.
- Hence image can be defined as a two-dimensional function, $f(x, y)$ where x and y are spatial coordinates and the amplitude of f at any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.
- When x , y and the intensity values of f are all finite discrete quantities we say the image is a digital image.
- A digital image is composed of a finite number of elements, each of which has a particular location and value.
- These elements are called image elements, picture elements, pels or pixels.
- Pixel is the term used most widely to denote the elements of a digital image.
- As we discuss earlier vision is the most advanced of our senses, therefore images play the single most important role in human perception system.

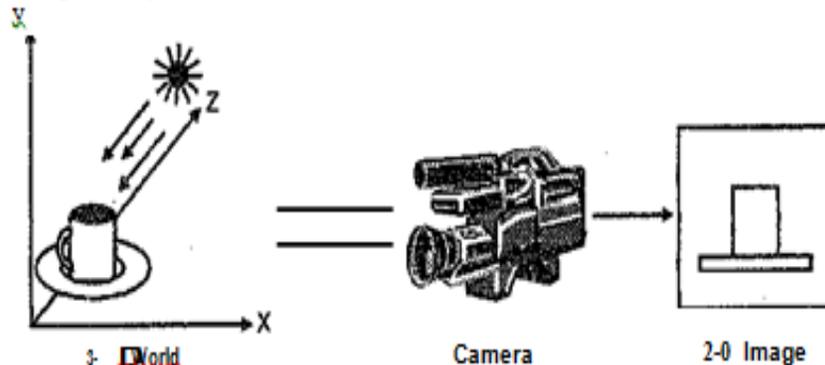


Fig.13.2.1

- In 2-dimension image we lose one dimension.
- Depth information is lost in this moving process from 3-dimension object to 2-dimension image as shown in Fig.13.2.1.
- Photographs on identity card, all family pictures etc. are example of 2-dimensional image.

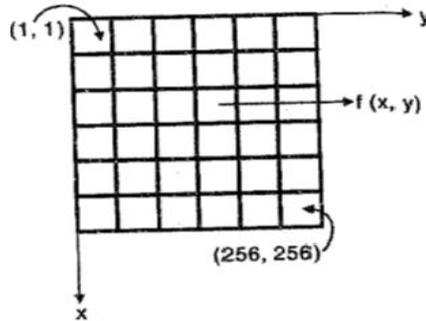


FIG 13.2.2

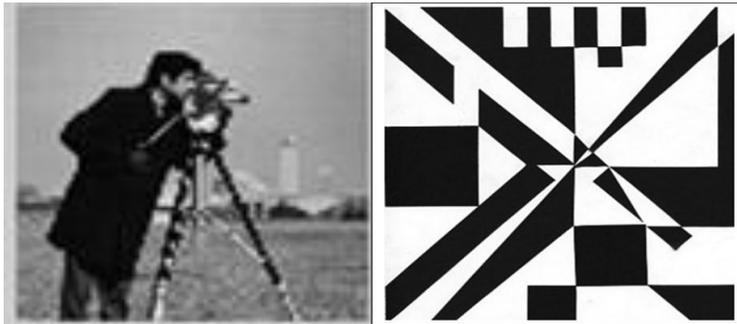


FIG 13.2.3

FIG 13.2.4

- In Fig.13.2.2, to locate the pixel we need to know its position in the two directions x and y .
- Here (x, y) are spatial coordinates and f is grey level (colour in case of colour images) at that point.
- Therefore all the images we see are 2-dimensional function.
- A typical image is represented as shown in Fig.13.2.3 and Fig.13.2.4.

13.2.1 Types of Image:

It was stated earlier images are 2-dimensional functions. Images can be classified as follows:

- (1) **Monochrome Image**
- (2) **Grey Scale Image**
- (3) **Colour Image**

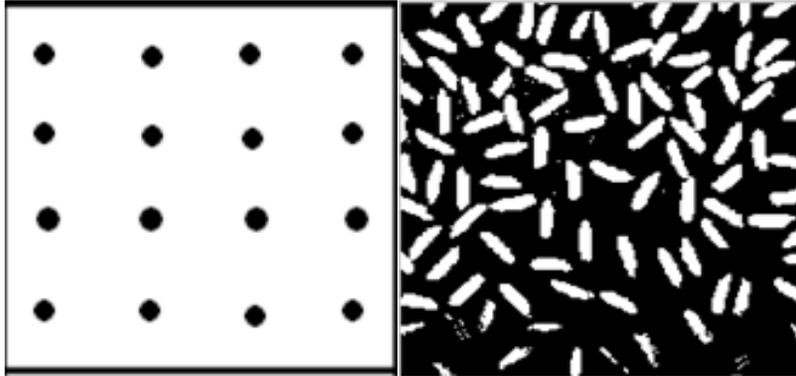


Fig. 13.2.1.1

Fig. 13.2.1.2

(1) Monochrome Image:

- In this type of image, each pixel is stored as a single bit 0 or 1.
- These images are also called as bit mapped image or binary image.
- Here 0 represent black and 1 represent white.
- It is also called as black and white image.
- In this image we have black and white pixels and no shades of grey as shown in Fig. 13.2.1.1 and Fig. 13.2.1.2.

(2) Grey Scale Image:

- In this type of image, each pixel is usually stored as 8-bits (1 byte).
- Because of this each pixel can have value ranging from 0 to 255.
- Here 0 represent black and 255 represent white.
- In this image we have black, white and various shades of grey present in image as shown in Fig. 13.2.1.3 and Fig 13.2.1.4.

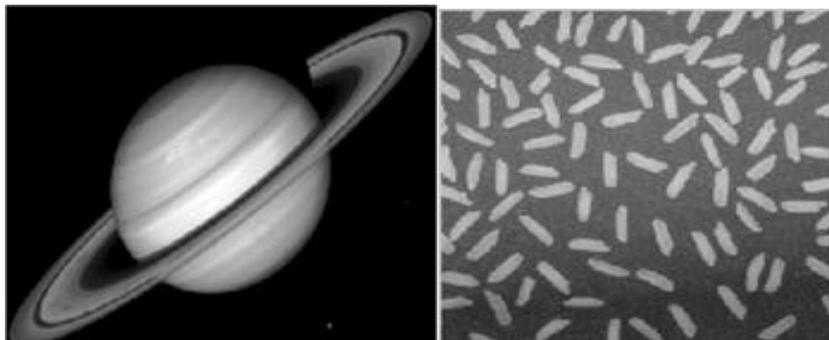


Fig. 13.2.1.3

Fig. 13.2.1.4

(3) Colour Image:

- In this type of image, each pixel is usually stored as 24-bits.
- Variety of colours can be generated by mixing the three primary colours Red, Green and Blue in appropriate proportions.
- In colour images, each pixel is composed of RGB values and each of colours requires 8-bits for its representation.

- Hence each pixel is represented by 24-bits[R (8 bits), G (8 bits),B(8 bits)].

Colour images can be easily converted to grey scale image using the equation,

$$\mathbf{X} = \mathbf{0.30R+0.59G+0.13B} \quad \dots (13.2.1.1)$$

Easier formula to achieve same result as follows,

$$\mathbf{X} = \mathbf{R + G + B / 3} \quad \dots (13.2.1.2)$$

13.3 DIGITAL IMAGE FILE FORMATS

- A file format is a structure which defines how information is stored in the file and how that information is displayed on the monitor.
- There are numerous image file formats which are available.
- Some of the image formats that can be used on either Macintosh or PC platforms are:
 1. BMP (Bit Mapped Graphic Image)
 2. JPEG (Joint Photographic Expert Group)
 3. TIFF (Tagged Image File Formal)
 4. EPS (Encapsulated Post Script)
 5. GIF (Graphic Interchange Format)
 6. PICT (Macintosh Supported)
 7. PNG (Portable Network Graphics)

1. BMP:

- Images developed by Microsoft can save both monochromes as well as colour images.
- All the wallpaper on computer is example of BMP images.
- When we work in paint Brush, we can save our work as BMP image.
- The quality of BMP files is very good but they occupy a lot of memory space.

2. JPEG:

- **Joint Photographic Expert Group** is the name of committee that developed this JPEG image format which is used in compression algorithms.
- JPEG images are compressed images which occupy very little space.

- It is based on the Discrete Cosine Transform-DCT.
- JPEG images use lossy compression algorithms which result in some loss of original data.
- The quality of JPEG images is not as good as BMP images.

3. TIFF:

- Most well-known formats were developed in 1986.
- A TIFF format is also a data compression technique for monochrome as well as colour images.
- Images seen on the Internet sites are normally TIFF/GIF images simply because they occupy less space.

4. EPS:

- It is file format of the post script page description language and is device independent.
- This means that images can readily be transferred from one application to another.
- However EPS image can only be printed in script compatible printers.

5. GIF:

- Uses a form of Huffman coding.
- Used in Internet images
- Limited color palette with RGB values, in the range of 0 to 255.
- CMYK (Cyan, Magenta, Yellow, and Key (black)) colors are not possible in GIF.

6. PICT:

- It is a general purpose format supported by Macintosh and used for storing bit-mapped images.

7. PNG:

- It is lossless image format.
- To improve and replace the GIF format, PNG image format was designed.
- The image file shows organisation of information encoded that consists of two parts:
 1. Header
 2. Image data.

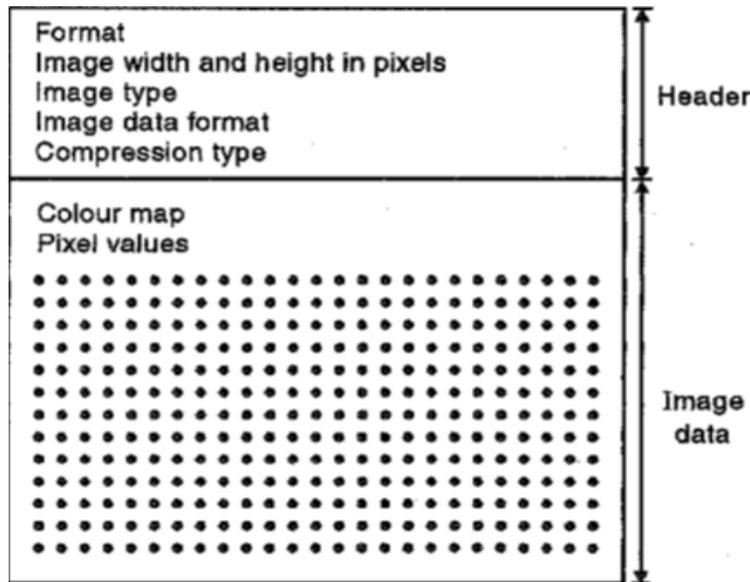


Fig.13.3.1

- **Header:** The header file gives us the information about the kind of image.
- **Format:** The header file identifies the format as binary code or ASCII string.
- **Width and Height:** The width and height of the image are given in number of pixels.

Common image types: **binary images, 8-bit greyscale images, 8-bit colour and 24-bit colour images.**

- **Image data format:** It specifies the order in which pixel values are stored in the image data section. Order used is left to right and top to bottom.
- Image data format also specifies if the RGB values in the image are interlaced or not interlaced.
- Example of interlaced: **R G B R G B R G B**
 1 pixel 2 pixel 3 pixel
- Example of not interlaced: **R R R.... G G G.... B B B....**
- By interlaced we mean that the RGB values of each pixel stay together consecutively followed by the three colour components for the next pixel.
- If the RGB values are not interlaced, the values of one primary colour for all pixels appear first, then the values of the next primary colour followed by the values of the third primary colour.
- **Compression type:** It shows image data is compressed.
- There are a few more formats given below:

1. PGM (Portable Grey Map): Represent grey level image.
2. PBM (Portable Bit Map): Represent binary images.
3. PPM (Portable Pixel Map): Represent colour images.

13.4 IMAGE COMPRESSION

- In comparison to text files, images take up a huge amount of disk space.
- The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information.
- Various amounts of data may be used to represent same amount of information.
- This is known as data redundancy.
- Data redundancy is central issue in digital image compression.
- Basically there are three basic redundancies:
 1. **Coding redundancy.**
 2. **Interpixel redundancy.**
 3. **Psychovisual redundancy.**
- Data compression is achieved when one or more of these redundancies are eliminated.
- In image compression based on methodologies, there are two types of compression as follows:
 - **Lossy compression:** Part of original data is discarded permanently to reduce file size. After decompression original data cannot be retrieved back.
Advantage: No degradation of quality.
 - **Lossless compression:** Original data is not changed permanently during compression. After decompression original data can be retrieved back.
Disadvantage: Degradation of quality.

13.4.1 IMAGE COMPRESSION STANDARD-JPEG:

- JPEG stands for Joint Photographic ExpertGroup.
- Image compression is known to most users of computers in the form of image file extensions such as the jpg file extension.
- JPEG images are compressed images which occupy very little space.
- It is based on the Discrete Cosine Transform (DCT).
- JPEG images use lossy compression algorithms.
- This compression is used to reduce the size of file without damaging its quality.

JPEG Compression Process:

- First, we convert the R, G, B color format to Y, Cb and Cr format.
- Some colors are more sensitive to human eye, thus are high-frequency color and some of chromium compounds like Cb and Cr are less sensitive thus can be ignored.
- Then we reduce the size of pixels by down sampling process.
- We divide our image into 8*8 pixels and perform forward DCT (Direct Cosine Transformation).
- Then we perform quantization using quantum tables.
- Finally we compress our data using various lossless compression methods such as run-length encoding, Huffman encoding etc.

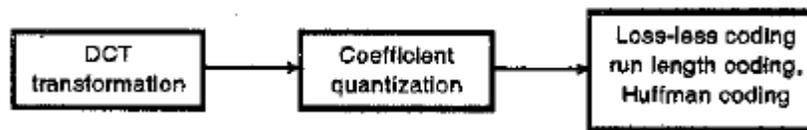


Fig.13.4.1.1

13.5 IMAGE PROCESSING

- Image Processing is a subject which basically studies the images and manipulation of images by computer.
- It involves processing the images that are obtained by camera.
- With the beginning of computers, image processing grew rapidly as a subject.
- It was stated earlier that human eye-brain mechanism represents ultimate imaging system.

Now let us understand this with a scenario.

What happens when we look at an object?

Let us take a simple example: when we see an explosion, we immediately identify it as something danger. Two things have happened here:

1. The scene has been captured by the eye.
2. The brain processed the scene and gave out a warning signal.

This is known as Image Processing.

- In above example the scene is captured by the eye and it sends the signal to the brain.
- Brains processed this signal and some meaningful information is generated.
- When working with computer, images captured from camera are fed into the system where algorithm is written to process these images.

- Hence in image processing system, camera replaces the human eye and computer replaces brain which does the processing.
- The field of digital image processing refers to processing digital images by means of a digital computer.

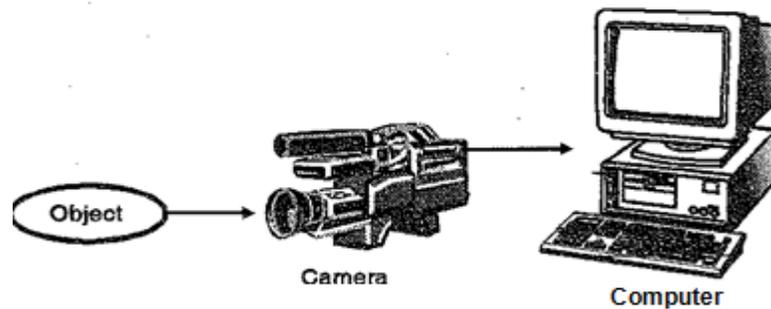


Fig.13.5.1

- Therefore we start processing the images the day we are born and continue till we die.
- Hence image processing is integral part of us.
- An another important trait that is common to all human being is to store the information, analyse it, discuss it with others and try to make it better.

13.5.1 DIGITAL IMAGE ENHANCEMENT:

- Image enhancement is the next step after image acquisition in fundamental image processing.
- There are many improvisation methods to manipulate an images and the result is more suitable than the original image.
- Here image is enhanced.
- The desired result can vary from person to person.
- In image enhancement no extra information is added to the original image.
- It simply improves the subjective quality of an image by working with the existing data of an image.
- Image enhancement can be carried in two domains:

1. Spatial domain

2. Frequency domain

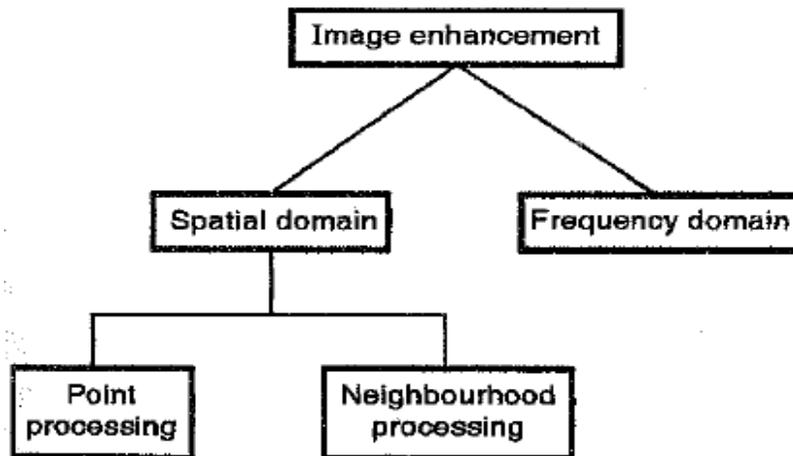


Fig: 13.5.1.1

1. Spatial Domain:

- Spatial domain means working in the given space.
- It implies working with the pixels values of an image.
- Assume $f(x, y)$ is original image. Here f is grey value and (x, y) are image coordinate.
- For 1 byte image f values range from 0-255, where 0 represent black, 255 represent white and all intermediate values represent shades of grey.
- The modified image can be expressed as,

$$g(x, y) = T[f(x, y)]$$

Where $g(x, y)$ is modified image.

$f(x, y)$ is original image.

T is the transformation applied on original image.

- Spatial domain enhancement can be carried out in two ways: Point processing and Neighbourhood processing.

2. Frequency Domain:

- Frequency domain means working with the frequencies value of an image.
- Frequency domain techniques are the study of working in the Fourier domain.
- Example: 1D-Fourier transforms, 2D-Fourier transform, 1D-DFT and 2D-DFT etc.

13.5.2 CONTRAST STRETCHING:

- We obtain low contrast images because many reasons such as poor illumination, lack of dynamic range in the imaging sensor or wrong setting of a lens aperture during image acquisition.

- Contrast stretching expands the range of intensity levels in an image.
- This means that contrast stretching increases dark portion to darker and bright portion to brighter.
- Contrast stretching is also used to increase the dynamic range of the modified image.
- Fig 13.5.2.1 shows transformation used to achieve contrast stretching.

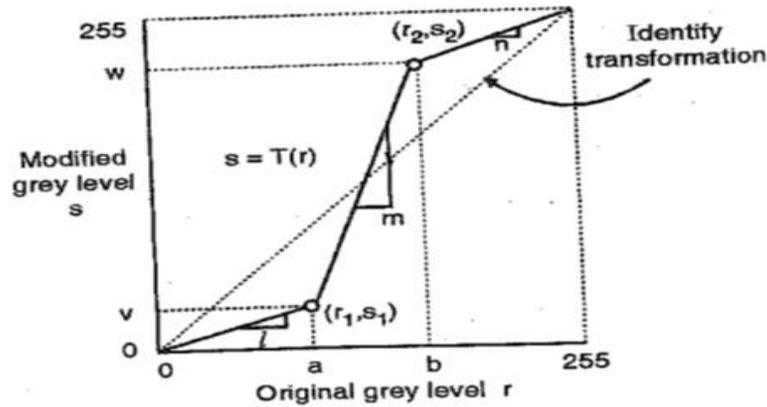


Fig. 13.5.2.1

- In figure dotted line indicate identity transformation and solid line indicate contrast stretching transformation.
- Here the dark grey levels are made darker by assigning slope value less than one and bright grey levels are made brighter by assigning slope value greater than one.
- The equation of the contrast stretching algorithm is given as:

$$\begin{aligned}
 g(x, y) &= l \cdot f(x, y) && \text{for } 0 \leq r < a && \text{or} \\
 g(x, y) &= m \cdot (r - a) + v && \text{for } a \leq r < b && \text{or} \\
 g(x, y) &= n \cdot (r - b) + w && \text{for } b \leq r < L-1
 \end{aligned}$$

where $g(x, y)$ is modified image.

$f(x, y)$ is original image.

l, m, n are slopes.

- It is clear from figure that l and n slope values are less than one while m slope is greater than one.
- Fig 13.5.2.2 shows original image and Fig 13.5.2.3 shows contrast stretched image.



Fig. 13.5.2.2

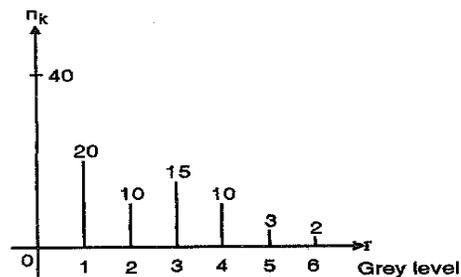


Fig. 13.5.2.3

13.5.3 HISTOGRAM:

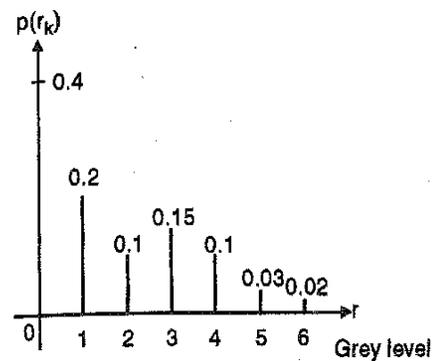
- Histogram of images provides a global description of the appearance of an image.
- The information obtained from histograms is enormous.
- Histogram of an image represents the relative frequency of occurrence of the various grey levels in an image.
- Histogram of an image can be plotted in two ways.
- In the first method, the x-axis has the grey levels and the y-axis has the number of pixels in each grey level.
- In the second method, the x-axis represents the grey levels, while the y-axis represents the probability of the occurrence of that grey level.

Grey level	Number of pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2



Method 1

Grey level	Number of pixels (n_k)	$p(r_k) = \frac{n_k}{n}$
0	40	0.4
1	20	0.2
2	10	0.1
3	15	0.15
4	10	0.1
5	3	0.03
6	2	0.02
$n = 100$		



Method 2

- In method 2,

$$P(r_k) = n_k / n$$

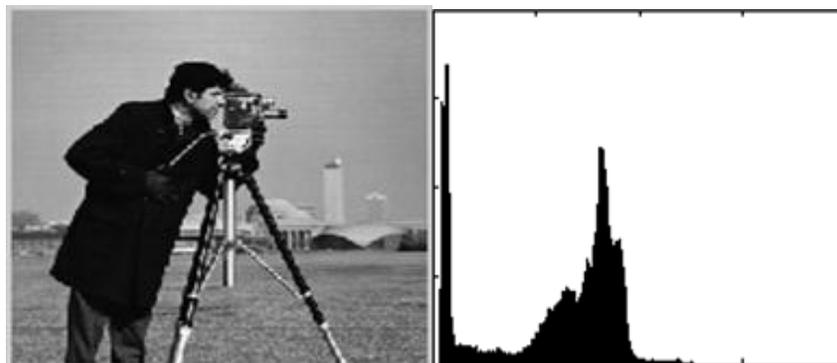
$r_k \rightarrow k^{\text{th}}$ grey level

$n_k \rightarrow$ Number of pixel in the k th grey level

$n \rightarrow$ Total number of pixel in an image

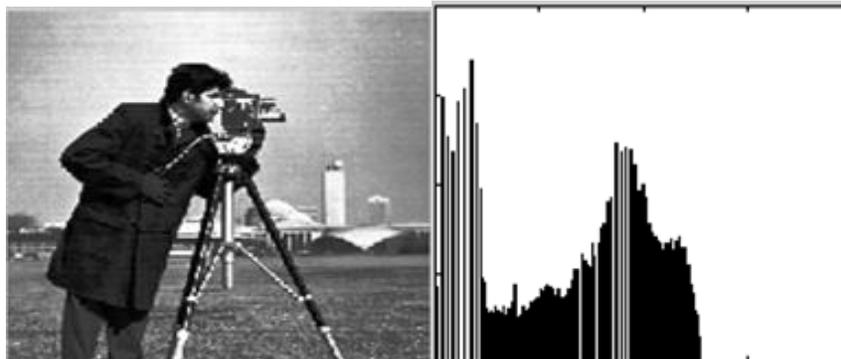
13.5.3.1 HISTOGRAM EQUILIZATION:

- There are many applications where we need flat histogram.
- Histogram equalization gives us flat histogram.
- A perfect image is one which has equal number of pixels in all its grey level, which is achieved by ideal histogram equalization.
- This technique carries the transformations that transform bad histogram to a flat histogram.
- It also increases the dynamic range of a given image along with equal distribution of pixels in its grey levels.
- The transformation we are looking for must satisfy the following two conditions:
 1. $T(r)$ must be single valued and monotonically increasing in the interval $0 \leq r \leq 1$.
 2. $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$ i.e. $0 \leq s \leq 1$ for $0 \leq r \leq 1$.
Here range of r is [0, 1]. This is called normalized range.



Original Image

Histogram



Equalized Image

Equalized Histogram

13.5.4 FILTERING:

- As we studied in image enhancement topic, neighbourhood processing is one of the important technique in spatial domain.
- In neighbourhood processing we consider pixel as well its immediate neighbours to enhance the image.

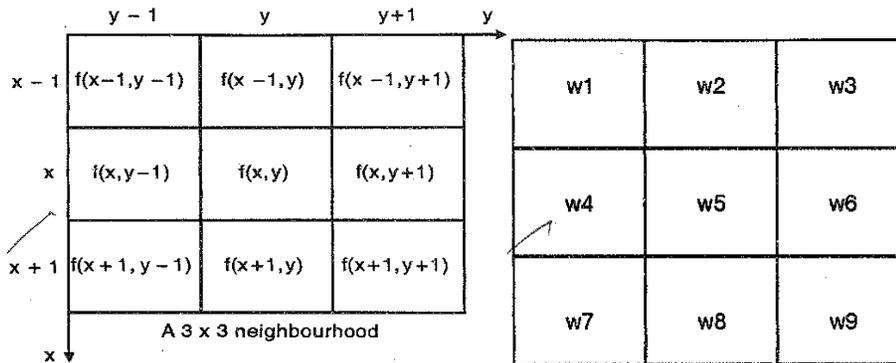


Fig. 13.5.4.1

Fig. 13.5.4.2

- Fig.13.5.4.2 shown is called filter, mask, window or a template.
- Mask can be 3 x 3, 5 x 5, 7 x 7 and so on.
- To process image we place mask on image, multiply each component of mask with corresponding value of image, add them up and place the calculated value at center.
- Consider $f(x, y)$ is the original image and $g(x, y)$ is modified image then,

$$g(x,y)=w1*f(x-1,y-1)+w2*f(x-1,y)+w3*f(x-1,y+1)+w4*f(x,y-1) \\ +w5*f(x,y)+w6*f(x,y+1)+w7*f(x+1,y-1)+w8*f(x+1,y) \\ +w9*f(x+1,y+1)$$

- Once $g(x, y)$ is calculated we shift the mask by one step towards the right to the next pixel.
- One of the important operations of neighbourhood processing is image filtering.
- In filtering we remove the noise from the image.
- In most images, the background is considered to be low frequency regions and edges are considered to be high frequency region.

13.5.4.1 SMOOTHING:

- Smoothing is also called as low pass filtering.
- Smoothing implies removing or blurring the edges.

- Smoothing is used to remove noise present in the image.
- Noise is normally high frequency component and low pass filtering removes these high frequency components.
- Here we are going to study smoothing in two ways: **Low Pass Averaging Filter and Median Filter.**

13.5.4.1.1 LOW PASS AVERAGE FILTER

- This filter is used to eliminate Gaussian noise from the image.
- Frequency response of low pass averaging filter along with spatial response is shown in figure.
- We generate the mask which gives us low pass filtering operation.
- The coefficient use in mask should be positive in case of low pass averaging filter.

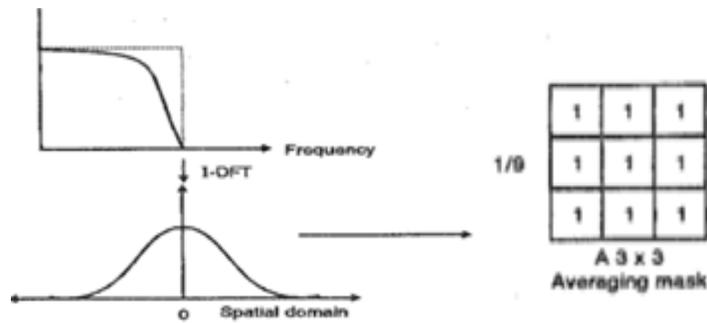


Fig.13.5.4.1.1.1

- Let us take the 3 x 3 average masks shown in Fig.13.5.4.1.1.1 on the pseudo image given below to understand working of low pass averaging filter.
- Consider the size of image is 8 x 8.
- It is clear that 8 x 8 image have single edge between 10 and 50.
- Place the 3 x 3 average masks on this image.
- Start from top left hand corner.
- Multiply each component of image with the corresponding value of mask.
- As all values of image inside mask are 10, we get average as 10 and center pixel remains 10.
- Therefore according to equation of neighbourhood processing,

$$\begin{aligned}
 g(x, y) &= 1*10+1*10+1*10+1*10+1*10+1*10+1*10+1*10+1*10/9 \\
 &= 10+10+10+10+10+10+10+10+10/9 \\
 &= 90/9 \\
 &= 10
 \end{aligned}$$

Next shift the mask toward right till we reach end of line, perform

same step of calculation as explain above and then move it downwards.

- Some of calculation of mask position is shown below.

10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	<u>10</u>	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50

10	<u>10</u>	10	10	10	10	10	10	10	10	<u>10</u>	10	10	10
10	10	<u>10</u>	10	10	10	10	10	10	10	10	<u>10</u>	10	10
10	10	10	<u>10</u>	10	10	10	10	10	10	10	10	<u>10</u>	10
10	10	10	10	<u>10</u>	10	10	10	10	10	10	10	10	<u>10</u>
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50

10	10	10	10	10	10	<u>10</u>	10	10	10	10	10	10	10
10	10	10	10	10	10	10	<u>10</u>	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	<u>10</u>	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50

10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	<u>10</u>	10	10	10	10	10	10	10
10	10	10	10	10	10	10	<u>10</u>	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	<u>10</u>	10	10	10	10
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50	50	50	50	50

10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
23.3	23.3	23.3	23.3	23.3	23.3	23.3	23.3	23.3	23.3
36.6	36.6	36.6	36.6	36.6	36.6	36.6	36.6	36.6	36.6
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50

- Once we get final result you can observe that the low frequency region has remained unchanged. Hence we can conclude that sharp edge has become blur.
- The sharp edge between 10 and 50 has become blurred.
- We can see from grey level 10 it changes to 23.3(after rounding off its 23), from 23 it changes to 36.6(after rounding off its 36) and finally from 36 it changes to 50.

13.5.4.1.2 LOW PASS MEDIAN FILTER:

- We have seen low pass averaging filter blurs the edges by eliminating the noise.
- Bigger the averaging mask more is blurring.
- Also there are chances image can contain salt and pepper noise.
- If we work with low pass averaging filter to remove the same it will blur noise but at same time it will also ruin the edges.
- Therefore we need to work with non-linear filter.
- Such non-linear filter is called as Median filter.
- Low pass median filter removes the salt and pepper noise from the image
- They are also called as order statistic filters because their response is based on ranking of the pixels contained within the mask.
- Consider using the mask similar to averaging filter except that the mask will contain no values.
- The step to perform median filtering is as follows:
 1. Assume 3 x 3 empty mask.
 2. Place the empty mask at top left hand corner of given pseudo image.
 3. Arrange all nine pixels inside mask in ascending or

descending order.

4. Choose the median value from these nine pixels and place at center.

5. Move the mask in similar fashion as done in averaging filter.

- Consider the 8 x 8 pseudo image with salt and pepper noise given below.
- We need to remove noise without disturbing the edges.
- Place the empty mask at top left hand corner.
- Arrange the nine pixels of image inside mask either in ascending or descending order as given below.

10 10 10 10 10 10 10 10 250

- The median is value placed at fifth position. Therefore this value is placed at the center.
- Continue this procedure until all salt and pepper noise get replaced by median value.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	<u>250</u>	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	<u>250</u>	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	<u>50</u>	50
50	50	50	50	50	50	50	50

<u>10</u>	10	10	10	10	10	10	10
10	<u>10</u>	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	250	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	<u>50</u>	50
50	50	50	50	50	50	50	50

- Hence we get the final result as salt and pepper noise got eliminated without distortion of edges.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

13.6 SUMMARY

- Image is 2 dimensional representation of 3 dimensional World.
- We can store different types of images such as binary image, grey scale image, 8-bits colour image, and 24 bits colour image.
- Image processing is a study in which images are recorded by camera and are processed by the computer.
- Various file formats are used such as jpeg, gif, bmp etc.
- Image data can be compressed and good image compression standard like jpeg is available.
- There are image enhancement methods like contrast stretching, histogram equalization and filtering which subjectively improvised and enhance the original image into more suitable one.

13.7 REFERENCES

- Digital Image Processing (Using Matlab Codes) by Dhananjay K. Theckedath.

13.8 BIBLIOGRAPHY

1. Digital Image Processing Fourth Edition Rafael C. Gonzalez and Richards E. Woods.

13.9 EXERCISE

1. Define Images. State and explain types of images.
2. Write a short note on histogram equalization.
3. Explain low pass averaging filter and low pass median filter.
4. Write a short note on image enhancement.
5. List and explain image file formats.
6. Explain image compression jpeg standards.
