# QUANTUM *Series*

**Semester - 5** | **CS & IT**

## Computer Graphics

- Topic-wise coverage of entire syllabus in Question-Answer form.

- Short Questions (2 Marks)

# QUANTUM SERIES

*For*
B.Tech Students of Third Year
of All Engineering Colleges Affiliated to
**Dr. A.P.J. Abdul Kalam Technical University,
Uttar Pradesh, Lucknow**
(Formerly Uttar Pradesh Technical University)

**Computer Graphics**

By
Kanika Dhama



**QUANTUM PAGE PVT. LTD.**
Ghaziabad ■ New Delhi

**Computer Graphics (CS : Sem-5)**

1st Edition : *2009-10*
2nd Edition : *2010-11*
3rd Edition : *2011-12*
4th Edition : *2012-13*
5th Edition : *2013-14*
6th Edition : *2014-15*
7th Edition : *2015-16*
8th Edition : *2016-17*
9th Edition : *2018-19* (*Thoroughly Revised Edition*)
10th Edition : *2020-21*

*Price: Rs. 80/- only*

*Printed Version :* e-Book.

# ■ CONTENTS ■

## KCS–053 : COMPUTER GRAPHICS

| Computer Graphics (KCS-053) | | |
|---|---|---|
| **Course Outcome ( CO)** | | **Bloom's Knowledge Level (KL)** |
| **At the end of course , the student will be able to:** | | |
| CO 1 | Understand the graphics hardware used in field of computer graphics. | $K_2$ |
| CO 2 | Understand the concept of graphics primitives such as lines and circle based on different algorithms. | $K_2$, $K_4$ |
| CO 3 | Apply the 2D graphics transformations, composite transformation and Clipping concepts. | $K_4$ |
| CO 4 | Apply the concepts of and techniques used in 3D computer graphics, including viewing transformations. | $K_2$, $K_3$ |
| CO 5 | Perform the concept of projections, curve and hidden surfaces in real life. | $K_2$, $K_3$ |
| **DETAILED SYLLABUS** | | **3-0-0** |
| **Unit** | **Topic** | **Proposed Lecture** |
| I | **Introduction and Line Generation:** Types of computer graphics, Graphic Displays- Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Mid-point circle generating algorithm, and parallel version of these algorithms. | **08** |
| II | **Transformations:** Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections and shearing. **Windowing and Clipping:** Viewing pipeline, Viewing transformations, 2-D Clipping algorithms- Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against non rectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping, Curve clipping, Text clipping | **08** |
| III | **Three Dimensional:** 3-D Geometric Primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping. | **08** |
| IV | **Curves and Surfaces:** Quadric surfaces, Spheres, Ellipsoid, Blobby objects, Introductory concepts of Spline, Bspline and Bezier curves and surfaces. | **08** |
| V | **Hidden Lines and Surfaces:** Back Face Detection algorithm, Depth buffer method, A- buffer method, Scan line method, basic illumination models– Ambient light, Diffuse reflection, Specular reflection and Phong model, Combined approach, Warn model, Intensity Attenuation, Color consideration, Transparency and Shadows. | **08** |

**Text books:**

1. Donald Hearn and M Pauline Baker, "Computer Graphics C Version", Pearson Education
2. Foley, Vandam, Feiner, Hughes – "Computer Graphics principle", Pearson Education.
3. Rogers, " Procedural Elements of Computer Graphics", McGraw Hill
4. W. M. Newman, R. F. Sproull – "Principles of Interactive computer Graphics" – Tata MCGraw Hill.
5. Amrendra N Sinha and Arun D Udai," Computer Graphics", Tata MCGraw Hill.
6. R.K. Maurya, "Computer Graphics " Wiley Dreamtech Publication.
7. Mukherjee, Fundamentals of Computer graphics & Multimedia, PHI Learning Private Limited.
8.  Donald Hearn and M Pauline Baker, "Computer Graphics with OpenGL", Pearson education

# 1
UNIT

# Introduction and Line Generation

# CONTENTS

---

### PART-1

*Introduction and Line Generation : Types of Computer Graphics.*

### CONCEPT OUTLINE

- Computer graphics is a branch of computer science that deals with the theory and techniques of computer image synthesis.
- Types of computer graphics :
  i. Passive computer graphics
  ii. Interactive computer graphics

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 1.1.** **What is computer graphics ? Explain important applications of computer graphics.**

**Answer**

1. Computer graphics is an art of drawing pictures on computer screens with the help of programming.
2. It involves computation, creation, manipulation of data.
3. Computer graphics is a rendering tool for the generation and manipulation of images.

**Applications of computer graphics :**

1. **Graphical User Interface (GUI) :** Computer graphics tools is used to make GUI.
2. **Computer arts :** Computer graphics are used in designing object shapes and specifying object such as cartoon drawing, logo design.
3. **Education and training :** Computer generated models of physical financial and economic systems are used as educational aids. Learning with visual aids is fast, easy to understand and cost effective.
4. **Entertainment :** Graphics and image processing techniques can be used to transform an object into another object.
5. **Visualization :** Visualization is used to convert large data value into patterns, charts, graphs, etc., with the help of computer graphics.
6. **Presentation graphics :** With the help of computer graphics large volumes of business data can be presented easily, making it attractive and useful.

**Que 1.2.** | **Discuss the various types of computer graphics.**

**Answer**

**Various types of computer graphics are :**

1. **Passive (off-line) computer graphics :** The most common example of passive computer graphics is static website, where user has no control over the contents on the monitor. In this, development takes place independently in off-line mode.

2. **Interactive computer graphics :** In interactive computer graphics, user can interact with the machine as per his requirements. Videogames, dynamic websites, special effects in movies, cartoons are all making use of interactive computer graphic.

**Computer graphics can be broadly divided into the following classes :**

1. Business graphics or the broader category of presentation graphics, which refers to graphics, such as bar-charts (also called histograms), pie charts, pictograms (*i.e.*, scaled symbols), *x-y* charts, etc., used to present quantitative information to inform and convince the audience.

2. Scientific graphics, such as *x-y* plots, curve fitting, contour plots, system or program flowcharts etc.

3. Scaled drawing, such as architectural representations, drawing of buildings, bridges, and machines.

4. Cartoons and artwork, including advertisements.

5. Graphical User Interfaces (GUIs) which are the images that appear on almost all computer screens these days, designed to help the user utilize the software without having to refer to manuals or read a lot of text on the monitor.

**Que 1.3.** | **List some advantages and disadvantages of interactive computer graphics.**

**Answer**

**Advantages of interactive computer graphics :**

1. It provides tools for producing pictures of concrete, 'real-world' objects.

2. It has an ability to show moving pictures, and thus it is possible to produce animations with interactive graphics.

**Disadvantages of interactive computer graphics :**

1. Requires technical skill to produce.

2. Must be updated daily to keep audience engaged.

---

**PART-2**

*Graphics Displays-Random Scan Displays, Raster Scan Displays.*

## CONCEPT OUTLINE

- Random scan display is a technique used for producing images on CRT.
- Raster scan display is a technique used for displaying images on CRT.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 1.4.** **Explain raster scan display and random scan display device.**

AKTU 2012-13, Marks 05

**OR**

**Differentiate between raster scan display and random scan display with example.**

**Answer**

**Raster scan display :**

1. In raster scan displays, screen is scanned in horizontal and vertical direction and the information is stored in a buffer called frame buffer.

2. The frame buffer is used to store intensity values of all screen points.

3. It is suitable for displaying realistic scenes containing either complex shades or colour patterns.

4. Simple black and white display require only one bit per pixel while colour display systems require multiple bits per pixel.

5. Refreshing on raster scan displays is carried out at the rate of 60 to 80 frames per second.

**Random scan display :**

1. In random scan display, the definition of picture is stored as a collection of line of commands in an area of memory called refresh buffer or display program.

2. Random scan display draw a picture on line at a time and for this reason is also referred to as vector displays.

3. It is basically designed for line drawing and not suitable for complex natural scenes.

4. It refreshes at a rate of 30 to 60 frames per second.

| S. No. | Raster scan display | Random scan display |
|--------|---------------------|---------------------|
| 1. | It is well suited for the realistic display of scenes containing subtle shading and colour patterns. | It is designed for line drawing applications and cannot display realistic shaded scenes. |
| 2. | Raster scan have low resolution than random system. | Random scan have higher resolution than raster system. |
| 3. | Picture definition is stored in form of pixel intensity value. | Picture definition is stored in form of line drawing algorithm. |
| 4. | It contains hidden surface techniques. | It does not contain hidden surface techniques. |
| 5. | The electron beam is swept across the screen, one row at a time, from top to bottom. | The electron beam is directed only to the parts of screen where a picture is to be drawn. |
| 6. | It is used for photos. | It is used for text, logs, letter heads. |
| 7. | Home television sets and dot-matrix printer is an example of raster scan system. | Pen plotter is an example of random scan system. |

**Que 1.5.** | **Explain the advantages of raster scan display over random scan display.**

**Answer**

**The advantages of raster scan display over random scan display :**
1. Less memory costs than random scan display.
2. High degree of realism achieved in picture than random scan display.
3. It uses advanced shading and hidden surface technique.
4. Computer monitors and TVs use this method.
5. Less expensive than vector display.
6. Very efficient to represent full images.

**Que 1.6.** | **What is interlacing ?**
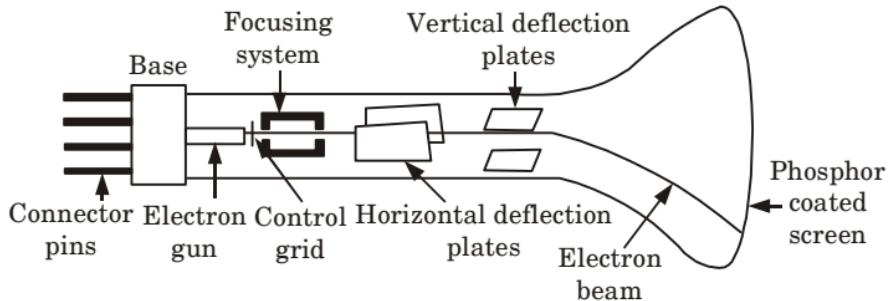
**Answer**
1. Interlacing is a method of encoding a bitmap image such that a person who has partially received it, sees a degraded copy of the entire image.
2. When communicating over a slow communications link, this is often preferable to seeing a perfectly clear copy of one part of the image, as it

helps the viewer to decide more quickly whether to abort or continue the transmission.

3. It is also known as interleaving.

**Que 1.7.** | **Explain the working of Cathode Ray Tube (CRT).**

**Answer**



**Fig. 1.7.1.** CRT.

**Working of CRT :**

1. CRT is an evacuated glass tube equipped with various components as shown in Fig. 1.7.1.

2. A beam of electrons (cathode rays), emitted by an electron gun, passes through focusing and deflection systems hits on the phosphor coated screen to generate the desired picture.

3. The high speed electrons hit the phosphor coated screen to produce a spot of light controlled by a video controller.

4. The electron gun in the CRT is made up of a heated metal cathode and a control grid.

5. The cathode is heated by passing a current through a coil of wire, called the filament.

6. The electrons get boiled off the hot cathode surface and move in the form of electron beams passing through horizontal and vertical deflection plates.

7. The negatively charged electrons are accelerated towards the phosphor coating by a high positive voltage.

8. The accelerating voltage can be generated with a positively charged metal coating on the inside of the CRT or by an accelerating anode.

9. Spots of light are produced when high speed electrons in the electron beam collide with the phosphor coating and their kinetic energy is absorbed by the phosphor.

10. Part of the beam energy is converted by friction into heat energy and the remaining energy is used to move electrons in the phosphor atom into higher energy levels.

11. After a short span of time, these excited electrons come back to their stable ground state and give up their extra energy as a small quantum of light energy.

12. The colour of the light emitted by these electrons is proportional to the energy difference between the excited quantum state and the ground state.

13. The CRT screen can be coated with different kinds (color and persistence) of phosphor.

14. The light emitted by the phosphor fades very rapidly. To maintain the picture on the display, we need to redraw the picture repeatedly by quickly directing the electron beam back over the same points such a type of display is known as refresh CRT.

---

**Que 1.8.** | **Differentiate between raster and vector graphics.**

**AKTU 2014-15, Marks 06**

**Answer**

| S. No. | Raster graphics | Vector graphics |
|--------|-----------------|-----------------|
| 1. | In raster graphics, the beam is moved all over the screen. | In vector graphics, the beam is moved between the end points of the graphics primitive. |
| 2. | Scan conversion is required. | Scan conversion is not required. |
| 3. | It also displays smooth lines by approximating them with pixels on the raster grid. | It draws a continuous and smooth line. |
| 4. | Cost is low. | Cost is more. |
| 5. | It has ability to display areas filled with solid colours or patterns. | It only draws lines and characters. |

---

**Que 1.9.** | **Discuss various emissive display devices.**

**OR**

**Write short note on :**
**i. Plasma panel**
**ii. LCD**

**Answer**

Various emissive display devices are :

**i.   Plasma panel :**

   a.   Plasma panels, also called gas discharge displays, are constructed by filling the region between two glass plates with a mixture of gases that usually includes neon.

   b.   A series of vertical conducting ribbons is placed on one glass panel and a set of horizontal ribbons is built into the other glass panel.

   c.   Firing voltages applied to a pair of horizontal and vertical conductors cause the gas at the intersection of the two conductors to break down into glowing plasma of electrons and ions.

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

   e.   Alternating current methods are used to provide faster application of the firing voltages, and thus brighter displays. Separation between pixels is provided by the electric field of the conductors.

**ii.   Liquid Crystal Displays (LCDs) :**

   a.   LCDs are non-emissive devices which produce a picture by passing polarized light from the surroundings of an internal light source through a liquid crystal material.

   b.   Liquid crystals are almost transparent substances, exhibiting the properties of both solid and liquid matter.

   c.   Light passing through liquid crystals change their molecular alignment and consequently the way light passed through them.

   d.   In LCDs, liquid crystals are sandwiched between thin polarized sheets that are used.

   e.   Liquid crystal displays are used in miniature televisions and video cameras and monitors.

---

**Que 1.10.**   **What are the flat panel displays devices ?**

**Answer**

1.   Flat panel display is a display method that is designed to reduce the depth of the CRT display caused by the length of the tube.

2.   The screens of these flat panel displays are made up of pairs of electrodes.

3.   Each pair of electrodes is used to generate one picture element.

4.   There are two types of flat panel displays :

   **a.   Emissive displays :**

      i.    The emissive displays (emitters) are devices that convert electrical energy into light.

      ii.   Plasma panels, light emitting diodes are examples of emissive displays.

**b. Non-emissive displays :**

i. Non-emissive displays (non-emitters) use optical effects to convert sunlight or light from some other source into graphics pattern.

ii. An example of a non-emissive flat panel display is a Liquid Crystal Device (LCD).

**Que 1.11.** **How advantageous are flat panel displays ? Explain the functioning of LCD with suitable diagrams.**

**Answer**

**Advantages of flat panel displays are :**

1. It consumes very low power.
2. It has very thin display which occupies small volume.
3. Cost is low.
4. Used for long lifetime.
5. Good reliability and brightness.

**Functioning of LCD :**

1. The principle behind the LCD is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist.

2. This changes the angle of light which is passing through the molecule of the polarized glass and also cause a change in the angle of the top polarizing filter.

3. As a result a little light is allowed to pass the polarized glass through a particular area of the LCD.

4. Thus, that particular area will become dark compared to other.

5. The LCD works on the principle of blocking light. While constructing the LCD, a reflected mirror is arranged at the back.

6. An electrode plane is made of Indium Tin Oxide (ITO) which is kept on top and a polarized glass with a polarizing film added on the bottom of the device.

7. The complete region of the LCD has to be enclosed by a common electrode and liquid crystal matter should be above it.

8. When there is no current, the light passes through the front of the LCD that will be reflected by the mirror and bounced back.

9. As the electrode is connected to a battery the current from it will cause the liquid crystals between the common plane electrode and the electrode shaped like a rectangle to untwist.

10. Thus, the light is blocked from passing through and that particular rectangular area appears blank.

Vertically Polarized Light

Liquid Crystal Molecules orientation

Horizontally Polarized Light

Un-polarized light

Light Output

ITO glass transparent electrodes

Horizontally Polarized Filter

Vertically Polarized Filter

Two filters are oriented in 90 degrees to each other
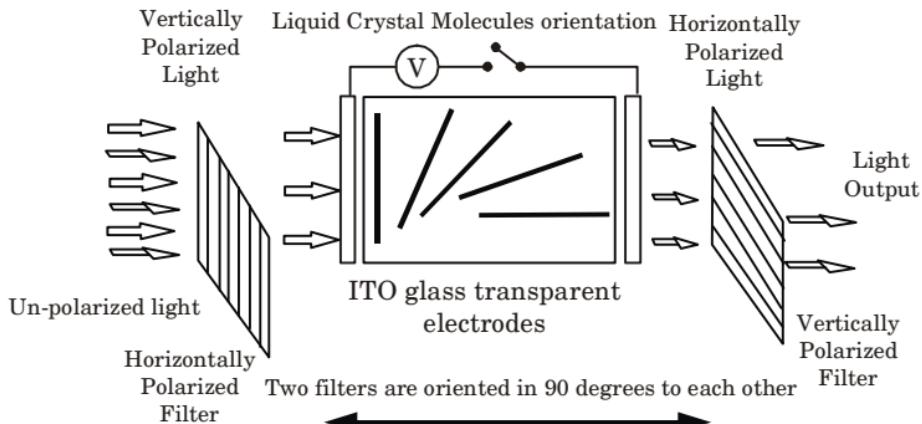
**Fig. 1.11.1.**

**Que 1.12.** **Write merits and demerits of LCD (Liquid Crystal Display).**                    AKTU 2014-15, Marks 06

**Answer**

**Merits of LCD :**

i.    It produces very bright images due to high peak intensity.

ii.   It produces lower electric, magnetic and electromagnetic fields than CRTs.

iii.  It has no geometric distortion at the native resolution. Minor distortion can occur for other resolutions.

iv.   It consumes less electricity than a CRT and produces little heat.

**Demerits of LCD :**

i.    The aspect ratio and resolution are fixed.

ii.   It is not good at producing black and very dark grays levels.

iii.  It has lower contrast than CRTs due to a poor black-level.

iv.   Images are satisfactory, but not accurate as colour saturation is reduced at low intensity levels due to a poor black-level.

PART-3

*Frame Buffer and Video Controller.*

CONCEPT OUTLINE

- Frame buffer is a memory area in which picture is stored in the form of a pixel.
- Video controller is used to control the operation of the display device.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 1.13.** Explain frame buffer and video basics. Also, explain the

terms pixel, aspect ratio, resolution. | **AKTU 2013-14, Marks 07**

**Answer**

**Frame buffer :**

1. Frame buffer is a special area of memory in raster displays which is dedicated to graphics.
2. It holds the set of intensity values for all the screen points.
3. The stored intensity values are retrieved from frame buffer and displayed on the screen one row (scanline) at the time.
4. Each screen point is referred to as a pixel or pel (shortened forms of picture element).
5. Each pixel on the screen can be specified by its row and column number. Thus, by specifying row and column number we can specify the pixel position on the screen.

**Video basics :**

1. Video or moving image in general, is created from a sequence of small images called frames.
2. By recording and then playing back frames in quick succession, an illusion of movement is created.
3. Video can be edited by removing some frames and combining sequences of frames, called clips, together in a timeline.

**Pixel :**

1. Pixel is the smallest part of the screen.
2. Each pixel has its own intensity, name or address by which we can control it.


Pixel

**Fig. 1.13.1.**

**Aspect ratio :**

1. An aspect ratio is an attribute that describes the relationship between the width and height of an image.

2. Aspect ratio is expressed by the symbolic notation *i.e.*, $x : y$.

**Resolution :**

1. Resolution is defined as the number of pixels on the horizontal axis and vertical axis.

2. The sharpness of the image on the display depends on the resolution and size of the monitor.

**Que 1.14.** **Write a short note on video controller and display processor.**

**Answer**

**Video controller :**

1. Video controller is a key hardware component that allows computers to generate graphic information to any video display devices, such as a monitor or projector.

2. They are also known as graphics or video adapters that are directly integrated into the computer motherboard.

3. Their main function as an integrated circuit in a video signal generator is to produce television video signals in computers systems.

4. They also offer various functions beyond accelerated image rendering, such as TV output and the ability to hook up to several monitors.

**Display processor :**

1. Display processor (or graphics controller) converts the digital information from the CPU to analog values needed by the display devices.

2. The purpose of the display processor is to free the CPU from the graphics work.

3. In addition to the system memory, a separate display processor memory area is provided.

4. A major task of the display processor is digitizing a picture definition given in an application program into a set of pixel values for storage in the frame buffer.

**Que 1.15.** **How much time is spent scanning across each row of pixels during a screen refresh of 1280 × 1024 at a refresh rate of 60 frames per second ?**

**Answer**

Resolution = 1280 × 1024.

Refresh rate = 60 frames/second.

Therefore, time taken by 1024 rows = 1/60 sec.
Hence, time taken by each row of pixels to refresh
$$= 1/(60 \times 1024) = 0.0162 \times 10^{-3} \text{ sec.}$$

**Que 1.16.** **Consider three different raster systems with resolutions 640 × 480, 1024 × 1280, and 2560 × 2048. What will be the size of frame buffer required for each of these systems to store 12 bits per pixel ? How much storage is required for full colours, that is, 24 bits per pixel ?**

**Answer**

Frame buffer size = (Number of pixel OR resolution × bits per pixel)/8 bytes.

| Resolution | 12-bit per pixel | 24-bit per pixel |
|------------|------------------|------------------|
| 640 × 480 | = (640 × 480 × 12)/8 | = (640 × 480 × 24)/8 |
| | = 460800 bytes | = 921600 bytes |
| 1024 × 1280 | = (1024 × 1280 × 12)/8 | = (1024 × 1280 × 24)/8 |
| | = 1966080 bytes | = 3932160 bytes |
| 2560 × 2048 | = (2560 × 2048 × 12)/8 | = (2560 × 2048 × 24)/8 |
| | = 7864320 bytes | = 15728640 bytes |

**Que 1.17.** **Consider two raster systems with resolutions of 640 × 480 and 1280 × 1024. How many pixels could be accessed per second in each of these systems by a display controller that refreshes the screen at a rate of 60 frames per second ?**

**AKTU 2015-16, Marks 10**

**Answer**

Number of pixel in one frame (resolution) = Number of column (width) × Number of row (height)
Number of frame in one second = 60 frames
So, number of pixel in 60 frames = resolution × 60
Hence,
a.  For resolution 640 × 480
    Number of pixels per second = 640 × 480 × 60 = 1843200 pixel
b.  For resolution 1024 × 1280
    Number of pixels per second = 1024 × 1280 × 60 = 78643200 pixel

**Que 1.18.** **Suppose we have a computer with 32-bits/word and a transfer rate of 1 mips (million instruction per second). How long would it take to fill the frame buffer of a 300 dpi (dot per inch) laser printer with page size of $8\frac{1}{2}$ inches by 11 inches.**

**Answer**

Frame buffer size = $8\frac{1}{2}$ × 11 (inches)$^2$ = $8\frac{1}{2}$ × 11 (300)$^2$

1 inch = 300 dpi = 8415000 dots = 8415000$n$ bits

say 1 dot = $n$ bits

Transfer rate = 1 mips

= $1 \times 10^6$ word per second    (1 mips = $10^6$ word)

= $32 \times 10^6$ bits/second (1 word = 32 bits)

Time required to fill the frame buffer = $\dfrac{8415000n}{32 \times 10^6}$ = 0.263 $n$ sec

If $n$ = 4 then time taken = 0.263 × 4 = 1.052 sec

**Que 1.19.** **If base address of a frame buffer is 100 and screen size is (15 inch × 19 inch) with resolution 13 dpi (dot per inch) calculate the memory location where the coordinate of pixels are stored.**

**i.  Pixel $P1$ at $A$ (200, 25)**

**ii. Pixel $P2$ at $B$ (75, 45)**                    | **AKTU 2014-15, Marks 06**

**Answer**

Base address = 100

Resolution of screen = 195 * 247 pixels (or dots)

**i.**  Memory location of $P1$ = Base address + (195 * 200 + 25) = 39125

**ii.** Memory location of $P2$ = Base address + (195 * 75 + 45) = 14770

---

## PART-4

*Points and Lines.*

---

### CONCEPT OUTLINE

- A point is a position in a plane. It has no size *i.e.* no width, no length and no depth. A point is shown by a dot.

- A line is defined as a line of points that extends infinitely in two directions. It has one dimension *i.e.*, length.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 1.20.** Define point and lines. Derive the equation for the intercept form of the line.

**Answer**

**Points and lines :**

1. A point is a position in a plane. It has no size *i.e.* no width, no length and no depth. A point is shown by a dot.

2. A line is defined as a line of points that extends infinitely in two directions. It has one dimension *i.e.*, length.

**Derivation :**

1. Line can be represented by two points *i.e.*, both the point will be on the line and lines are also described by the equation. Any point which satisfies the equation is on the line.

2. If two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ specify a line and another third point $P(x, y)$ also satisfy the equations then the slope between points $P_1P$ and $PP_2$ will be equal.

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$
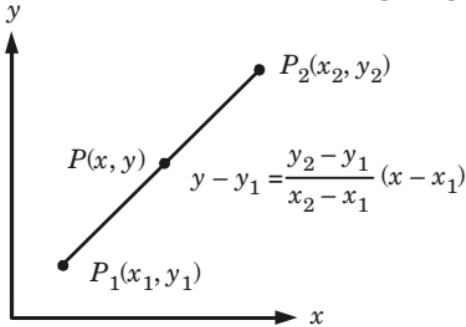
$$(y - y_1)(x_2 - x_1) = (y_2 - y_1)(x - x_1)$$

$$y(x_2 - x_1) = (y_2 - y_1)(x - x_1) + y_1(x_2 - x_1)$$

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x - x_1) + y_1 \qquad ...(1.20.1)$$

3. This is the equation of a line passing through the points $(x_1, y_1)$ and $(x_2, y_2)$ because $(x_1, y_1)$ and $(x_2, y_2)$ are numerical values so $\frac{y_2 - y_1}{x_2 - x_1}$ will be constant, let its value be $m$

Then, equation (1.20.1) becomes $y = m(x - x_1) + y_1$



**Fig. 1.20.1.**

$$y - y_1 = m(x - x_1)$$

where $$m = \frac{y_2 - y_1}{x_2 - x_1}$$
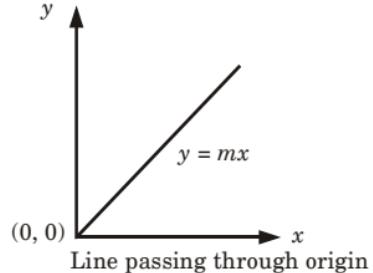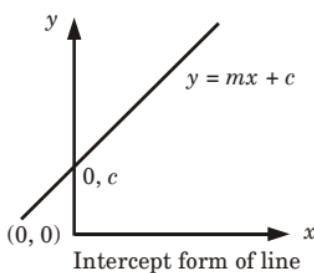
$$y - y_1 = mx - mx_1$$
$$y = mx - mx_1 + y_1$$
$$y = mx + (-mx_1 + y_1)$$

Where value $(-mx_1 + y_1)$ is constant. Let its value be $c$

$$y = mx + c \qquad \qquad ...(1.20.2)$$

4. If line $y = mx + c$ is passing through origin $(0, 0)$, then $c = 0$ and putting the value of $c$ in equation (1.20.2), we get

$$y = mx \qquad \qquad ...(1.20.3)$$



**Fig. 1.20.2.**

Equation (1.20.3) is a line passing through origin. Now from equation (1.20.1) we have

$$(y_2 - y_1)x - (x_2 - x_1)y + x_2y_1 - x_1y_2 = 0$$

Let $$y_2 - y_1 = A, \ x_2 - x_1 = -B, \ x_2y_1 - x_1y_2 = C$$

$$Ax + By + C = 0$$

This is the general form of the line.

---

**Que 1.21.** | **Derive the condition for which two lines are perpendicular or parallel.**

**Answer**

**Angle between two lines :** Let there are two lines $y = m_1x + c_1$ and $y = m_2x + c_2$ having tangent $m_1 = \tan\theta_1$ and $m_2 = \tan\theta_2$

$$\theta = \theta_1 - \theta_2$$
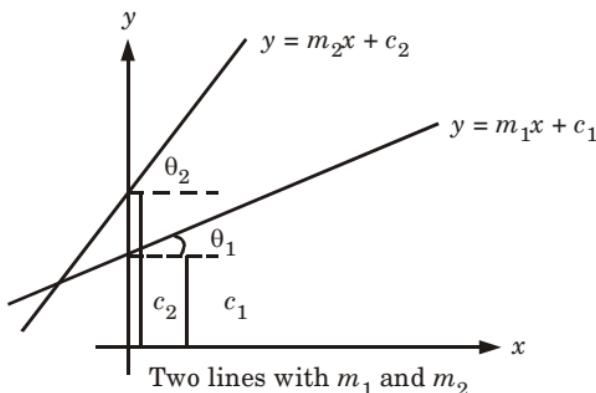
(Since $m_1 > m_2$, $\theta_1 - \theta_2$ otherwise $\theta_2 - \theta_1$)

Taking tan both sides

$$\tan\theta = \tan(\theta_1 - \theta_2)$$
$$= \left(\frac{\tan\theta_1 - \tan\theta_2}{1 + \tan\theta_1 \cdot \tan\theta_2}\right) = \left(\frac{m_1 - m_2}{1 + m_1 \cdot m_2}\right)$$
$$\theta = \tan^{-1}\left(\frac{m_1 - m_2}{1 + m_1 \cdot m_2}\right)$$

**Fig. 1.21.1.**

If lines are $a_1x + b_1y + c_1 = 0$ and $a_2x + b_2y + c_2 = 0$ then putting the value of $m_1 = -a_1/b_1$ and $m_2 = -a_2/b_2$, we get

$$\theta = \tan^{-1}\left(\frac{b_1a_2 - a_1b_2}{a_1a_2 + b_1b_2}\right)$$

**Case 1 :** If both line are parallel then angle should be zero.

$$\tan^{-1}\left(\frac{m_1 - m_2}{1 + m_1 \cdot m_2}\right) = 0$$

$$\frac{m_1 - m_2}{1 + m_1 \cdot m_2} = 0$$

$$m_1 - m_2 = 0$$

$$m_1 = m_2 \ i.e., \text{ slopes are equal.}$$

Similarly,         $a_1b_2 = b_1a_2$

**Case 2 :** If both are perpendicular then angle should be 90°.

$$\tan^{-1}\left(\frac{m_1 - m_2}{1 + m_1 \cdot m_2}\right) = \frac{\pi}{2}$$

$$\frac{m_1 - m_2}{1 + m_1 \cdot m_2} = \tan\left(\frac{\pi}{2}\right) = \infty$$

$$1 + m_1 \cdot m_2 = 0$$

$$m_1 \cdot m_2 = -1$$

or    $\tan^{-1}\left(\dfrac{a_2b_1 - a_1b_2}{a_1a_2 + b_1b_2}\right) = \dfrac{\pi}{2}$

$$a_1a_2 + b_1b_2 = 0$$

## PART-5

*Line Drawing Algorithm.*

## CONCEPT OUTLINE

- There are two line generation algorithm :
  i. Digital Differential Analyzer (DDA)
  ii. Bresenham's line algorithm.
- DDA is a scan conversion line algorithm based on calculating either $\Delta x$ or $\Delta y$.
- Bresenham's line algorithm is a scan conversion process for lines with positive slope less than 1.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 1.22.** Describe DDA line drawing algorithm.

**OR**

Implement the DDA algorithm to draw a line from (0, 0) to (6, 6).

**OR**

Explain symmetrical DDA with suitable example.

**Answer**

**DDA algorithm :**

1. Read the line endpoint $(x_1, y_1)$ and $(x_2, y_2)$

2. $$\Delta x = |x_2 - x_1|$$
   $$\Delta y = |y_2 - y_1|$$

3. If $(\Delta x \geq \Delta y)$ then

   $$\text{length} = \Delta x$$

   else

   $$\text{length} = \Delta y$$

4. Select the raster unit, *i.e.*,

   $$\Delta x = \frac{(x_2 - x_1)}{\text{length}}$$

   $$\Delta y = \frac{(y_2 - y_1)}{\text{length}}$$

5. $$x = x_1 + 0.5 * \text{Sign} (\Delta x)$$
   $$y = y_1 + 0.5 * \text{Sign} (\Delta y)$$

   Where sign function makes the algorithm work in all quadrants. It returns – 1, 0, 1 depending on whether its agreement is < 0, = 0, > 0 respectively.

The factor 0.5 makes it possible to round the values in the integer function rather than truncating them.

Now plot the points

```
i = 1
while (i ≤ length)
{
Plot (integer (x), integer (y))
x = x + Δx
y = y + Δx
i = i + 1
}
```

6.  Stop.

**Numerical :** Compute initial values :

$$\Delta x = x_2 - x_1 = 6 - 0 = 6$$

$$\Delta y = y_2 - y_1 = 6 - 0 = 6$$

$$m = \frac{\Delta y}{\Delta x} = 1$$

$$\text{length} = \Delta x$$

$$\Delta x = \frac{(x_2 - x_1)}{\text{length}}$$

$$\Delta x = \frac{6 - 0}{6}$$

$$\Delta x = \frac{6}{6}$$

$$\Delta x = 1$$

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

| Step | $x$ | $y$ | Pixel |
|------|-----|-----|--------|
| 0 | 0 | 0 | (0, 0) |
| 1 | 1 | 1 | (1, 1) |
| 2 | 2 | 2 | (2, 2) |
| 3 | 3 | 3 | (3, 3) |
| 4 | 4 | 4 | (4, 4) |
| 5 | 5 | 5 | (5, 5) |
| 6 | 6 | 6 | (6, 6) |



Pixel position
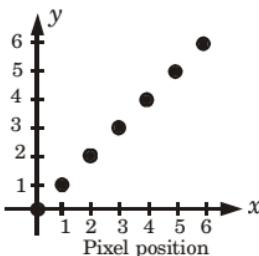
**Fig. 1.22.1.**

**Que 1.23.** Calculate value of pixels which is scan converted by DDA algorithm and endpoint of line is (4, 4) and (– 3, 0).

**Answer**

$$(x_1, y_1) = (-3, 0) \quad , \quad (x_2, y_2) = (4, 4)$$
$$\Delta x = 4 - (-3) = 7$$
$$\Delta y = 4 - 0 = 4$$
$$\text{Length} = \Delta x \qquad\qquad (\because \Delta x > \Delta y)$$
$$\Delta x = \frac{4 - (-3)}{7} = 1$$
$$\Delta y = \frac{4 - 0}{7} = \frac{4}{7} = 0.57$$
$$x = x_1 + 0.5 \times \text{sign}(\Delta x) = -3 + 0.5 \times 1 = -2.5$$
$$y = y_1 + 0.5 \times \text{sign}(\Delta y)$$
$$= 0 + 0.5 \times 0.57 = 0.28$$

Plot integer (– 2.5) and (0.28), *i.e.*, (– 3, 0)

$$x = x + \Delta x = -2.5 + 1 = -1.5$$
$$y = 0.28 + 0.57 = 0.85$$

Plot integer (– 1.5) and (0.85), *i.e.*, (– 2, 1)

$$x = -1.5 + 1 = -0.5$$
$$y = 0.85 + 0.57 = 1.42$$

Plot integer (– 0.5) and (1.42), *i.e.*, (– 1, 1)

$$x = -0.5 + 1 = 0.5$$
$$y = 1.42 + 0.57 = 1.99$$



**Fig. 1.23.1.**

Plot integer (0.5) and (1.99), *i.e.*, (1, 2)

$$x = 0.5 + 1 = 1.5$$
$$y = 1.99 + 0.57 = 2.56$$

Plot integer (1.5) and (2.56), *i.e.*, (2, 3)

$$x = 1 + 1.5 = 2.5$$
$$y = 2.56 + 0.57 = 3.13$$

Plot integer (2.5) and (3.13), *i.e.*, (3, 3)

$$x = 2.5 + 1 = 3.5$$
$$y = 3.13 + 0.57 = 3.7$$

Plot integer (3.5) and (3.7), *i.e.*, (4, 4)

**Que 1.24.** | **Write the Bresenham's algorithm of line and explain.**

**AKTU 2014-15, Marks 06**

**OR**

**Write steps required to draw a line from point $(x_1, y_1)$ to $(x_2, y_2)$ using Bresenham's line drawing algorithm.**

**OR**

**Describe Bresenham's line drawing algorithm. For 10 × 10 frame buffer, interpret the Bresenham's algorithm by hand to find which pixels are turned on for the line segment (1, 2) and (7, 6).**

**Answer**

**Bresenham's line algorithm :**

1. Input the two lines endpoints and store the left endpoint $(x_0, y_0)$.

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

3. Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for decision parameter as :

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

4. At each $x_k$ along the line, starting at $k = 0$, perform the following test :

   a. If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and

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

   b. If $p_k > 0$, the next point to plot is $(x_k + 1, y_k + 1)$ and
   $$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 $\Delta x$ times.

   **Solution of the given problem :**
   $$\Delta x = x_2 - x_1 = 7 - 1 = 6$$
   $$\Delta y = y_2 - y_1 = 6 - 2 = 4$$
   $$\text{Incr } c_1 = 2\Delta y = 8$$
   $$\text{Incr } c_2 = 2(\Delta y - \Delta x) = 2(4 - 6) = -4$$
   $$p_0 = 2\Delta y - \Delta x = 8 - 6 = 2$$

| Step | $P_k$ | Pixel |
|------|-------|-------|
|      | 2     | (2, 3) |
| 1    | – 2   | (3, 3) |
| 2    | 6     | (4, 4) |
| 3    | 2     | (5, 5) |
| 4    | – 2   | (6, 5) |
| 5    | 6     | (7, 6) |
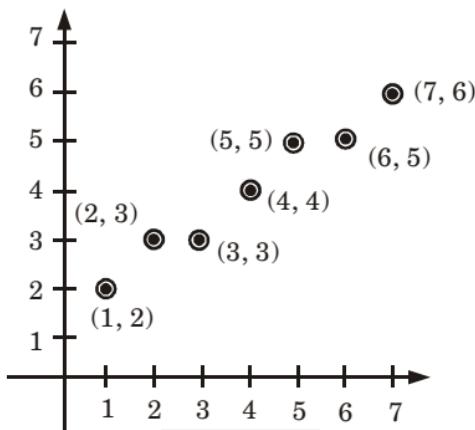


**Fig. 1.24.1.**

---

**Que 1.25.** Compare digital differential analyzer with Bresenham's line drawing algorithm.

AKTU 2015-16, Marks 10

**Answer**

| S. No. | Digital Differential Analyzer (DDA) line drawing | Bresenham's line drawing Algorithm |
|--------|--------------------------------------------------|-------------------------------------|
| 1. | DDA algorithm uses floating points *i.e.*, real arithmetic. | Bresenham's algorithm uses fixed points *i.e.*, integer arithmetic. |
| 2. | DDA algorithm uses multiplication and division in its operations. | Bresenham's algorithm uses subtraction and addition in its operations. |
| 3. | DDA algorithm is slowly than Bresenham's algorithm in line drawing. | Bresenham's algorithm is faster than DDA algorithm in line drawing. |
| 4. | DDA algorithm round off the coordinates to integer that is nearest to the line. | Bresenham's algorithm does not round off but takes the incremental value in its operation. |

**Que 1.26.** Develop and write Bresenham's algorithm of line. Also, predict the pixels on the line from (2, 2) to (12, 10) using this algorithm.

AKTU 2012-13, Marks 05

**Answer**

**Bresenham's line drawing algorithm :** Refer Q.1.24, Page 1–21C, Unit-1.

**Numerical :**

$$x_1 = 2, y_1 = 2, x_2 = 12, y_2 = 10$$
$$\Delta x = x_2 - x_1 = 12 - 2 = 10$$
$$\Delta y = y_2 - y_1 = 10 - 2 = 8$$

Slope of the line    $m = (y_2 - y_1) / (x_2 - x_1) = (10 - 2) / (12 - 2) = 0.8$

The initial decision parameter $p_0 = 2\Delta y - \Delta x$

$$p_0 = 2 \times 8 - 10 = 6$$

The increments for calculating successive decision parameters are $2\Delta y - 2\Delta x = 16 - 20 = -4$ and $2\Delta y = 16$, plot the initial point $(x_0, y_0) = (2, 2)$, successive are :

| Step | $p_k$ | Pixel |
|------|-------|-------|
| 0 | 6 | (3, 3) |
| 1 | 2 | (4, 4) |
| 2 | – 2 | (5, 4) |
| 3 | 14 | (6, 5) |
| 4 | 10 | (7, 6) |
| 5 | 6 | (8, 7) |
| 6 | 2 | (9, 8) |
| 7 | – 2 | (10, 8) |
| 8 | 14 | (11, 9) |
| 9 | 10 | (12, 10) |
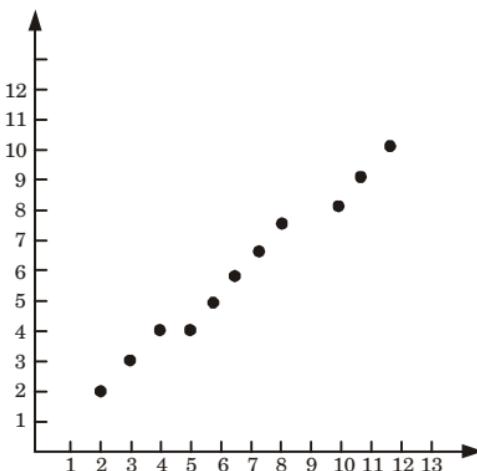
**Fig. 1.26.1.**

**Que 1.27.** **What are the criteria that should be satisfied by a good line drawing algorithm ? Explain.**    **AKTU 2015-16, Marks 10**

**Answer**

**Criteria for good line drawing algorithm :**

1. **Lines should appear straight :**

   a. Lines generated parallel to the *x*-axis or *y*-axis or at 45° are plotted correctly with point plotting techniques.

   b. Line may not pass through other addressable points in between.

2. **Lines should terminate accurately :**

   a. If lines are not plotted accurately, they may terminate at wrong places.

   b. This may lead to a small gap between endpoints of one line and the starting point of the next as shown in Fig. 1.27.1.



**Fig. 1.27.1.**

3. **Line should have constant density :**

   a. To maintain the constant density throughout the line, dots should be equally spaced.

b.  The line density is proportional to the number of dots displayed divided by the length of the line.

c.  Dots are equally spaced only in lines parallel to the *x*-axis or *y*-axis or at 45° to the axis as given in Fig. 1.27.2.



**Fig. 1.27.2.**

**4.  Line brightness independent of slope and line length :** This requires a high resolution of the device along with a high refresh rate.

**5.  Lines should be drawn rapidly :** In interactive applications, lines should appear rapidly on the screen, that is, minimum computation is desired to draw the line.

PART-6

*Circle Generating Algorithm, Mid-point Circle Generation Algorithm and Parallel Version of these Algorithm.*

CONCEPT OUTLINE

• Mid-point circle algorithm is used to draw a circle.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

Que 1.28. **Explain mid-point circle algorithm.**

AKTU 2014-15, Marks 07

**OR**

**Explain the mid-point circle generation algorithm.**

AKTU 2015-16, Marks 10

**Answer**

**Mid-point circle algorithm :**

1.  Input radius $r$ and center $(x_c, y_c)$ and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$ *i.e.*, initialize starting position as

$$x = 0$$
$$y = r$$

2.  Calculate the initial value of the decision parameter as

$$p = 1.25 - r$$

3.  do
```
{
        plot (x, y)
        if (p < 0)
            {
                x = x + 1
                y = y
                p = p + 2x + 1
            }
        else
            {
                x = x + 1
                y = y - 1
                p = p + 2x - 2y + 1
            }
        }
    while (x < y)
```

4.  Determine symmetry points in the other seven octants.

5.  Stop.

**Que 1.29.** Develop and write mid-point circle algorithm. Apply it to predict the pixels in any octant for the circle whose centre is origin and radius = 14 units.                    **AKTU 2013-14, Marks 07**

**Answer**

**Mid-point circle algorithm :** Refer Q.1.28, Page 1–25C, Unit-1.

**Numerical :**

The initial point $(x, y) = (0, 14)$

*i.e.,* $\qquad\qquad x = 0 \quad \text{and} \quad y = 10$

Calculate initial decision parameters $p = 1 - r$

$$= 1 - 14 = -13$$

$$p = -13 (< 0)$$

First plot $(0, 14)$

Here $p < 0$ so $\qquad\qquad x = 0 + 1 + 1$

$$y = 14$$

and $\qquad\qquad p = -13 + 2 \times 1 + 1 = -13 + 3 = -10$

$x < y$ *i.e.,* $1 < 14$ so, condition is true plot $(1, 14)$

Now $\qquad\qquad p = -10 (< 0)$

So $\quad x = \qquad\qquad 1 + 1 = 2$

$$y = 14$$

and $\qquad\qquad p = -10 + 2 \times 2 + 1 = -10 + 5 = -5$

$x < y$ *i.e.,* $2 < 14$ so, plot $(2, 14)$

Now $\qquad\qquad p = -5 (< 0)$

So $\qquad\qquad x = 2 + 1 = 3$

$$y = 14$$

$$p = -5 + 2 \times 3 + 1 = -5 + 7 = 2$$

$x < y$ so plot $(3, 14)$

Now $\qquad\qquad p = 2 (> 0)$

So $\qquad\qquad x = x + 1 = 3 + 1 = 4$

$$y = y - 1 = 14 - 1 = 13$$

$$p = p + 2x - 2y + 1$$

$$= 2 + 2 \times 4 - 2 \times 13 + 1$$

$$= 2 + 8 - 26 + 1$$

$$= 10 - 26 + 1 = -15$$

$x < y,$ so the next point to plot is $(4, 13)$

Now $\qquad\qquad p = -15 (< 0)$

$$x = x + 1 = 4 + 1 = 5$$

$$y = 13$$
$$p = p + 2x + 1$$
$$= -15 + 10 + 1 = -4$$



**Fig. 1.29.1.**

$x < y$, so next point to plot is $(5, 13)$

Now                     $p = -4 (< 0)$

So                      $x = x + 1$
$$= 5 + 1 = 6$$
$$y = 13$$
$$p = p + 2x + 1$$
$$= -4 + 12 + 1 = 8 + 1 = 9$$

$x < y$, so plot point $(6, 13)$

Now                     $p = 9 (> 0)$

So                      $x = x + 1 = 6 + 1 = 7$
$$y = y - 1 = 12$$
$$p = 9 + 14 - 24 + 1$$
$$= 9 - 10 + 1 = 0$$

**Note :** We have to continue iteration till $x = y$ or $x > y$.

**Que 1.30.** | **Using mid-point circle algorithm plot a circle with radius $r = 10$ units centred at origin.**

**Answer**

$$r = 10$$

The initial point $(x, y) = (0, 10)$

$$x = 0$$
$$y = 10$$

Calculate initial decision parameters $p = 1 - r = 1 - 10 = -9$

$$p = -9 \, (< 0)$$

First plot $(0, 10)$

Here $p < 0$ so            $x = 0 + 1 = 1$
$$y = 10$$

and            $p = -9 + 2 \times 1 + 1 = -9 + 3 = -6$

$x < y$ i.e., $1 < 10$. So condition is true and plot $(1, 10)$

Now            $p = -6 \, (< 0)$

So            $x = 1 + 1 = 2$
$$y = 10$$

and            $p = -6 + 2 \times 2 + 1 = -6 + 5 = -1$

$x < y$ i.e., $2 < 10$. So plot $(2, 10)$

Now            $p = -1 \, (< 0)$

So            $x = 2 + 1 = 3$
$$y = 10$$

and            $p = -1 + 2 \times 3 + 1 = 6$

$x < y$ so plot $(3, 10)$

Now            $p = 6 \, (> 0)$

So            $x = x + 1 = 3 + 1 = 4$
$$y = y - 1 = 10 - 1 = 9$$

and            $p = p + 2x - 2y + 1 = 6 + 8 - 18 + 1 = -3$

$x < y$ so plot $(4, 9)$

Now            $p = -3 \, (> 0)$

So            $x = x + 1 = 4 + 1 = 5$
$$y = y = 9$$

and            $p = p + 2x + 1 = -3 + 10 + 1 = 8$

$x < y$ so plot $(5, 9)$

Now            $p = 8 \, (> 0)$

So            $x = x + 1 = 5 + 1 = 6$
$$y = y - 1 = 9 - 1 = 8$$

and            $p = p + 2x - 2y + 1 = 8 + 12 - 16 + 1 = 5$

$x < y$ so plot (6, 8)

Now $\qquad\qquad$ $p = 5 \ (> 0)$

So $\qquad\qquad$ $x = x + 1 = 6 + 1 = 7$

$\qquad\qquad\qquad$ $y = y - 1 = 8 - 1 = 7$

and $\qquad\qquad$ $p = p + 2x - 2y + 1 = -5 + 14 - 14 + 1 = 6$

Plot (7, 7), now stop as $x$ equal to $y$.
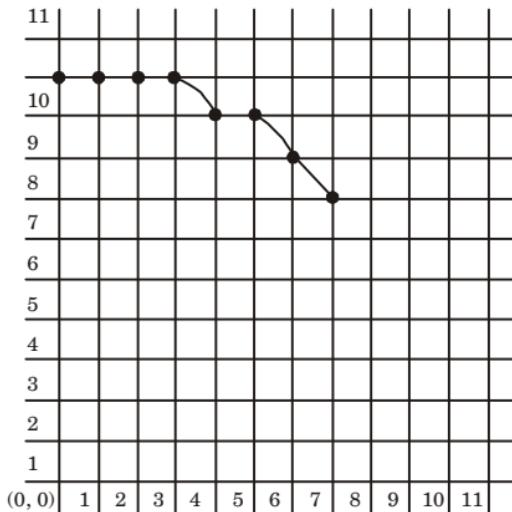


**Fig. 1.30.1.**

---

**Que 1.31.** | **Explain the parallel version of line algorithm by two methods.** | **AKTU 2012-13, 2013-14; Marks 07**

**Answer**

In the parallel version of line algorithm, pixel positions are calculated along a line path simultaneously by partitioning the computations among the various processors available. Methods for parallel version of line algorithm are :

**1. Bresenham's line algorithm :**

    a.   Given $n_p$ processors, a parallel Bresenham's line algorithm can be used by subdividing the line path into $n_p$ partitions and simultaneously generating line segments in each of the sub-intervals.

    b.   For a line with slope $0 < m < 1$ and left endpoint coordinate $(x_0, y_0)$, the line is partitioned along the positive $x$-direction.

c. The distance between beginning $x$ positions of adjacent partitions is calculated as :

$$\Delta x_p = \frac{\Delta x + n_p - 1}{n_p}$$

where $\Delta x$ is the width of the line and the value for partition width $\Delta x_p$ is computed using integer division.

d. The starting $x$ coordinate is calculated for the $k$th partition as :

$$x_k = x_0 + k\Delta x_p$$

e. At the $k$th partition, the starting $y$ coordinate is :

$$y_k = y_0 + \text{round}\ (k\Delta y_p)$$

f. The initial decision parameter for Bresenham's algorithm at the start of the $k$th sub interval is :

$$p_k = (k\Delta x_p)\ (2\Delta y) - \text{round}\ (k\Delta y_p)\ (2\Delta x) + 2\Delta y - \Delta x$$

**2. In raster scan system :**

a. In raster systems we assign each processor to a particular group of screen pixel.

b. For lines with a positive slope greater than 1, we reverse the value of $x$ and $y$. *i.e.*, we sample at unit $y$ intervals ($\Delta y = 1$) and calculate each succeeding $x$ value as

$$x_{k+1} = x_k + \frac{1}{m}$$

c. Lines are to be processed from the left endpoint to the right endpoint. If this processing is reversed, so that the starting endpoint is at the right, then either we have $\Delta x = -1$ and

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

or (when the slope is greater than 1) we have $\Delta y = -1$ with

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

d. Perpendicular distance $d$ from the line in to pixel as shown in Fig. 1.31.1 with coordinates $(x, y)$ is obtained with the calculation

$$d = Ax + By + C$$

where, $\quad A = \dfrac{-\Delta y}{\text{linelength}}$

$$B = \frac{\Delta x}{\text{linelength}}$$

$$C = \frac{x_0\Delta y - y_0\Delta x}{\text{linelength}}$$

$$\text{linelength} = \sqrt{\Delta x^2 + \Delta y^2}$$
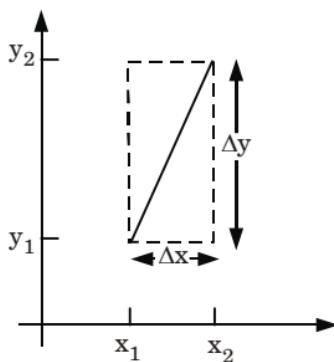


**Fig. 1.31.1.**

☺☺☺

# 2
UNIT

# Transformation

# CONTENTS

---

<div align="center">

## PART-1

*Transformations : Basic Transformation.*

</div>

<div align="center">

### CONCEPT OUTLINE

</div>

- Transformation is the process by which we can change the shape, position and direction of any object with respect to any coordinates system.
- Homogeneous coordinates allows us to express all transformation equations as matrix multiplication.

---

<div align="center">

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

</div>

---

**Que 2.1.** | **What are the basic transformations ? Describe each with their matrix representation.**

**Answer**

Transformations are the process by which we can change position, orientation or size of any coordinate system. Following are the various basic transformations :
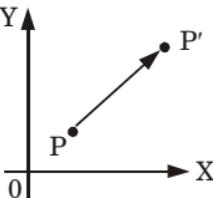
**1. Translation :**

    a. Translation is a process of changing the position of an object in a straight line path from one coordinate location to another.

    b. We can translate a two dimensional point by adding translation distances, $t_x$ and $t_y$, to the original coordinate position $(x, y)$ to move the point to a new position $(x', y')$, as shown in the Fig. 2.1.1.

$$x' = x + t_x \qquad \qquad ...(2.1.1)$$
$$y' = y + t_y \qquad \qquad ...(2.1.2)$$

    c. The translation distance pair $(t_x, t_y)$ is called translation vector.

    d. Translation equation (2.1.1) and (2.1.2) can be represented in the form of matrix as :



**Fig. 2.1.1.**

$$P = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
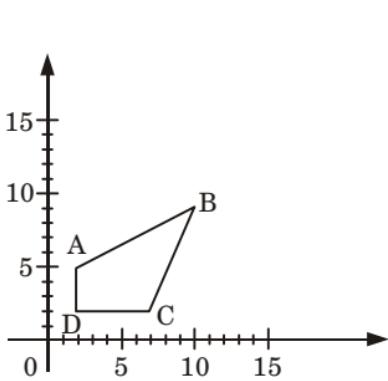
$$P' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix}$$

$$T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$
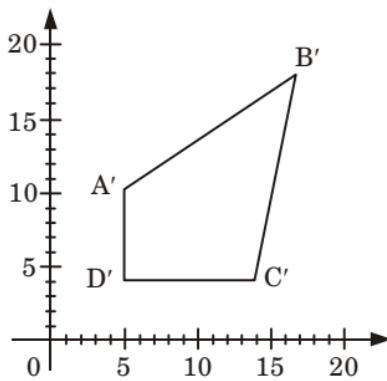
Hence $\qquad P' = P + T$

**2.  Scaling :**

a.  A scaling transformation changes the size of an object.



(a) Before scaling                          (b) After scaling

**Fig. 2.1.1.**

b.  Scaling operation is carried out for polygon by multiplying the coordinate values $(x, y)$ of each vertex by scaling factor $s_x$ and $s_y$ to produce the transformed coordinates $(x', y')$ as

$$x' = x \cdot s_x$$
$$y' = y \cdot s_y$$
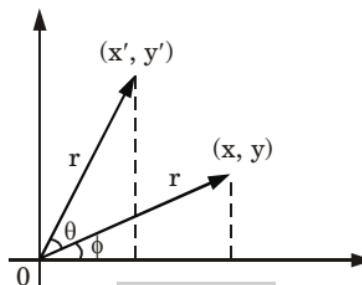
c.  Scaling factor $s_x$ scales objects in the $x$-direction, while $s_y$ scales objects in the $y$-direction.

d.  The transformation equation can also be written in the matrix form as :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

e.  If the value of scaling factor is less than 1 then it reduces the size of objects otherwise it increases the size of object.

**3. Rotation :**

a. Two dimensional rotation is applied to an object by rotating it along a circular path in the *xy* plane.

b. To generate a rotation, we specify a rotation angle θ and the position $(x_r, y_r)$ of the rotation point about which the object is to be rotated.

c. Positive values for the rotation angle define counterclockwise rotations about the pivot point and negative values rotate object in clockwise direction.



**Fig. 2.1.2.**

Using standard trigonometric identities, we can express the transformed coordinates in terms of angle θ and φ as :

$$x' = r \cos (\phi + \theta) = r \cos \phi . \cos \theta - r \sin \phi . \sin \theta$$

$$y' = r \sin (\phi + \theta) = r \cos \phi . \sin \theta + r \sin \phi . \cos \theta \quad ...(2.1.3)$$

The original coordinates of the point in polar coordinates are given as :

$$x = r \cos \phi \text{ and } y = r \sin \phi \qquad ...(2.1.4)$$

Putting equation (2.1.4) in equation (2.1.3), we get

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

We can write the rotation equation in matrix form as :

$$P' = PR$$

Where the rotation matrix is

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

**Que 2.2.** | **What do you understand by homogeneous coordinate ?**

**Answer**

1. Homogeneous coordinates are defined as the coordinates that represents all the geometric transformation equations as matrix multiplication.

2. Coordinates are represented with three element column vectors, and transformation operations are written as $3 \times 3$ matrices. For translation we have :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Which we can write in the abbreviated form as :

$$P' = T(t_x, t_y) \cdot P$$

3. The inverse of translation matrix is obtained by replacing the translation parameters $t_x$ and $t_y$ with their negatives i.e., $-t_x$ and $-t_y$.

4. Rotation transformation equations about the coordinate origin are now written as :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

or as            $P' = R(\theta) \cdot P$

We get the inverse of rotation matrix when $\theta$ is replaced with $-\theta$.

5. Finally a scaling transformation relative to the coordinate origin is now expressed as matrix multiplication.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

or as            $P' = S(s_x, s_y) \cdot P$

Replacing these parameters with their multiplicative inverse ($1/s_x$ and $1/s_y$) yields the inverse scaling matrix.

PART-2

*Composite Transformation.*

CONCEPT OUTLINE

• Transformation that involves more than one basic transformation is called composite transformation.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.3. | What do you mean by composite transformation ? Explain with example.

Answer

1. A composite transformation is two or more transformations performed one after the other.

2. If a transformation of the plane $T_1$ is followed by a second plane transformation $T_2$, then the result itself may be represented by a single transformation $T$ which is the composition of $T_1$ and $T_2$ taken in that order and written as :

$$T = T_1.T_2$$

3. Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.

4. A combined matrix is given by :

$$[T][X] = [X][T_1][T_2][T_3][T_4]....[T_n]$$

5. The change in the order of transformation would lead to different results, as in general matrix multiplication is not commutative, *i.e.*, $[A].[B] \neq [B].[A]$ and the order of multiplication.

**For example :** Let us perform a counterclockwise 45° rotation of triangle $A(2, 3)$, $B(5, 5)$, $C(4, 3)$ about point $(1, 1)$ using composite transformation.

We know that the overall transformation matrix for a counterclockwise rotation by an angle θ about the point $(x_p, y_p)$ is given as :

$$T_1 . R . T_2 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_p\cos\theta + y_p\sin\theta + x_p & -x_p\sin\theta - y_p\cos\theta + y_p & 1 \end{bmatrix}$$

Here, $\theta = 45°$, $x_p = 1$ and $y_p = 1$. Substituting values we get

$$T_1 . R . T_2 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1 & -\sqrt{2}+1 & 1 \end{bmatrix}$$

$$\therefore \quad \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \\ 5 & 5 & 1 \\ 4 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1 & -\sqrt{2}+1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\dfrac{1}{\sqrt{2}}+1 & \dfrac{3}{\sqrt{2}}+1 & 1 \\ 1 & \dfrac{8}{\sqrt{2}}+1 & 1 \\ \dfrac{1}{\sqrt{2}}+1 & \dfrac{5}{\sqrt{2}}+1 & 1 \end{bmatrix}$$

---

**Que 2.4.** **What do you understand by the term "Concatenation of transformations" ? What are its advantages ? If A and B are two different transformations, illustrate with suitable example that $A \cdot B \neq B \cdot A$.** | **AKTU 2014-15, Marks 07**

**Answer**

**Concatenation of transformation :** Refer Q. 2.3, Page 2–6C, Unit-2.

**Advantages of concatenation of transformation :**

More complex geometric and coordinate transformation are formed through process of concatenation of function.

**Example :**

$$A = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h & k & 1 \end{pmatrix}$$

$$A \cdot B = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h & k & 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ h & k & 1 \end{pmatrix}$$

$$B \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h & k & 1 \end{pmatrix} \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ hs & ks & 1 \end{pmatrix}$$

$$A \cdot B \neq B \cdot A$$

**Que 2.5.** **Show that uniform scaling and rotation make a commutative pair, but in general scaling and rotation are not commutative.**

**Answer**

For a two dimensional transformation :

General scaling, $S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Rotation, $R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

According to commutative property,

$$SR = RS$$

Now, Taking L.H.S

$$SR = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} s_x \cos \theta & -s_x \sin \theta & 0 \\ s_y \sin \theta & s_y \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Taking R.H.S

$$RS = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} s_x \cos\theta & -s_y \sin\theta & 0 \\ s_x \sin\theta & s_y \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using general scaling, commutative property is not satisfied.

When $s_x$ and $s_y$ are assigned the same value, a uniform scaling is produced.

Let                              $s_x = s_y = 1$

then                             $SR = RS$

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

Hence, commutative property is satisfied for uniform scaling.

---

### PART-3

*Reflections and Shearing.*

---

### CONCEPT OUTLINE

- Let line $L(y = mx + b)$ have a $y$-intercept $(0, b)$ and an angle $\theta$ with respect to $x$-axis.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 2.6.** Derive reflection matrices for reflection about the *x*-axis.

**OR**
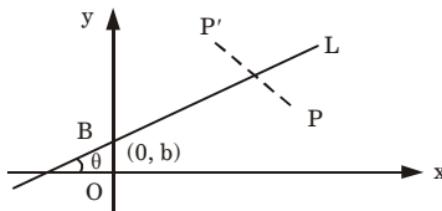
Establish the reflection transformation matrix about the line $y = mx + b$.                                    **AKTU 2012-13, Marks 10**

**Answer**

1. Let line $L(y = mx + b)$ have a $y$-intercept $(0, b)$ and an angle $\theta$ with respect to $x$-axis.

2. The steps involved in reflection transformation are as follows :

   a. Translate the intersection point $B$ to the origin.

b. Rotate by $-\theta°$ so that the line $L$ aligns with the $x$-axis.

c. Mirror reflect about the $x$-axis.

d. Rotate back by $\theta°$.

e. Translate $B$ back to $(0, b)$.



**Fig. 2.6.1.** Reflection of point $P$ about line $L$.

Therefore, we have

$$M_L = T_V \cdot R_\theta \cdot M_x \cdot R_{-\theta} \cdot T_{-V}$$

Where                          $V = bJ$

$$T_V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{-V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{-\theta} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Slope,                          $m = \tan\theta$

Therefore,            $\sin\theta = \dfrac{m}{\sqrt{m^2 + 1}}$

$$\cos \theta = \frac{1}{\sqrt{m^2 + 1}}$$

After multiplication of all basic matrix, we get

$$M_L = \begin{bmatrix} \dfrac{1-m^2}{m^2+1} & \dfrac{2m}{m^2+1} & \dfrac{-2bm}{m^2+1} \\[3mm] \dfrac{2m}{m^2+1} & \dfrac{m^2-1}{m^2+1} & \dfrac{2b}{m^2+1} \\[3mm] 0 & 0 & 1 \end{bmatrix}$$

**Que 2.7.** Write a procedure for rotation. Derive reflection metrics for reflection about *X*-axis.

AKTU 2015-16, Marks 10

**Answer**

**Procedure for rotation :** Rotation can be done in the following order :

**Step 1 :** Translate the object or body at the origin (translation).

**Step 2 :** Rotate by any angle as given (rotation).

**Step 3 :** Translate back to its original location (inverse translation).

In matrix form, it can be shown as

$$[T] = [T_R] \ [R_\theta] \ [T_R]^{-1}$$

where 
$$T_R \rightarrow \text{Translation matrix}$$
$$R_\theta \rightarrow \text{Rotation matrix by an angle } \theta°$$
$$[T_R]^{-1} \rightarrow \text{Inverse translation matrix}$$

**Reflection metrics :** For reflection about *x*-axis, *x* coordinate is not changed and sign of *y* coordinate is changed. Thus if we reflect point $(x, y)$ in the *x*-axis, we get $(x, -y)$

*i.e.*,
$$x' = x$$
$$y' = -y$$

So, the transformation matrix for reflection about *x*-axis or $y = 0$ axis is,

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

and the transformation is represented as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

**Que 2.8.** Show that the reflection about the line $y = -x$ is equivalent to a reflection relative to the $y$-axis followed by an anticlockwise rotation of 90°.

**Answer**

Reflection matrix about the line $y = -x$

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

Reflection matrix about the $y$-axis $\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Rotation matrix $\begin{pmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Reflection then rotation composite matrix is

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which is equal to the transformation matrix for reflection about the line $y = -x$.

**Que 2.9.** Reflect the polygon whose vertices are (–1, 0), (0, –2), (1, 0) and (0, 2) about the :

i. Horizontal line $Y = 2$

ii. Vertical line $X = 2$         **AKTU 2013-14, Marks 06**

**Answer**

Polygon is represented by homogeneous coordinates matrix as :

$$V = [ABCD]\begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}$$

**i.** **Horizontal line $Y = 2$ :** $Y = 2$

On comparing with $y = mx + c$

$$m = 0, c = 2$$

$$\theta = \tan^{-1}(m) = \tan^{-1}(0) = 0$$

The operation of reflection through an arbitrary line (not passing through origin) is required to follow the following procedures :

$$[T] = [T_{trans}] \, [R_\theta][R_{ref}][R_\theta]^{-1}[T_{trans}]^{-1}$$

$[T_{trans}]$ = Translation matrix

$[R_\theta]$ = Rotation matrix

$[R_{ref}]$ = Reflection matrix

$[R_\theta]^{-1}$ = Inverse rotation matrix

$[T_{trans}]^{-1}$ = Inverse translation matrix

$$[T_{trans}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -c & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 0 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos(\theta) & \sin(-\theta) & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{ref} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[R_\theta]^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_{trans}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & c & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

Reflected polygon = $\begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 2 & 0 \\ 0 & 4 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Now coordinates of polygon is, $(-1, 2)$, $(0, 4)$, $(1, 2)$ and $(0, 0)$.

**ii. Vertical line $X = 2$ :**

$$X = 2$$
$$y = mx + c$$

On comparing with $x = \dfrac{y}{m} - \dfrac{c}{m}$

$$m = \infty, c = -2$$
$$\theta = \tan^{-1}(m) = \tan^{-1}(\infty) = 90°$$

$$[T_{\text{trans}}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$

$$R_{\theta} = \begin{bmatrix} \cos(-90°) & \sin(-90°) & 0 \\ -\sin(-90°) & \cos(-90) & 0 \\ & & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{\text{ref}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_{\text{trans}}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

$$[R_{\theta}]^{-1} = \begin{bmatrix} \cos 90° & \sin 90° & 0 \\ -\sin(-90°) & \cos(-90°) & 0 \\ & & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T] = [T_{\text{trans}}][R_{\theta}][R_{\text{ref}}][R_{\theta}]^{-1}[T_{\text{trans}}]^{-1}$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\text{Reflected polygon} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & 2 & 0 \end{bmatrix}$$

Hence, new coordinates of polygon is,

(0, 1), (– 2, 2), (0, 3) and (2, 2).

**Que 2.10.** **What is shearing transformation ? Explain with examples.**

**Answer**
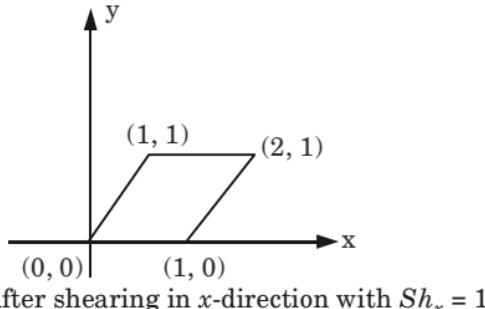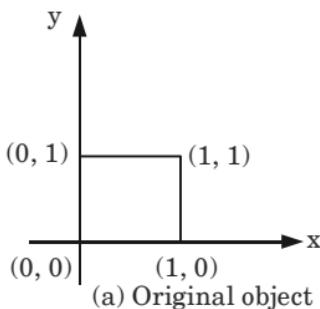
1. A transformation that slants the shape of an object is called the shearing transformation.

2. Shearing can be done :

   **a. Along x-direction :**

   i. An x-direction shear preserves the y coordinates, but changes the x coordinates which causes vertical lines to tilt right or left.

   ii. The transformation matrix for x shear is given as :

   $$Sh_x = \begin{bmatrix} 1 & Sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   $$x' = x + Sh_x \cdot y$$
   $$y' = y$$

   iii. Any real number can be assigned to the shear parameter $Sh_x$.



(a) Original object     (b) After shearing in x-direction with $Sh_x = 1$

**Fig. 2.10.1.**

**b. Along *y*-direction :**

i.   A *y*-direction shear preserve the *x* coordinates but changes the *y* coordinates which cause horizontal line to transform into lines which slope up or down.

ii.  The transformation matrix for *y*-shear is given as :

$$Sh_y = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x$$

$$y' = x.Sh_y + y$$



(a) Original object          (b) After shearing in *y*-direction with $Sh_y = 1$
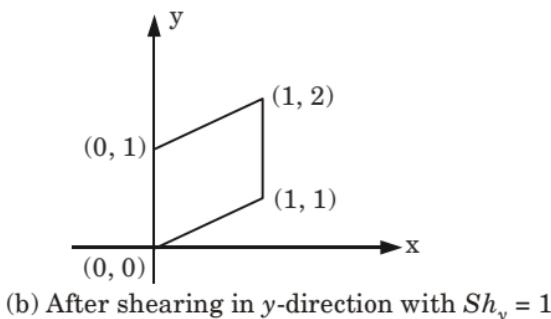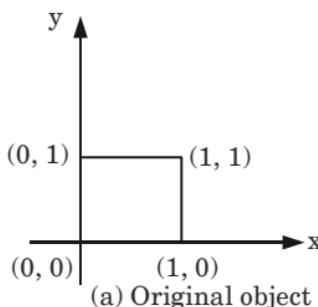
**Fig. 2.10.2.**

**c. Shearing with other reference line :**

i.   An *x*-direction shear relative to the line $y = y_{ref}$ is generated with transformation matrix given as :

$$Sh_{xref} = \begin{bmatrix} 1 & Sh_x & -Sh_x.y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x + Sh_x(y - y_{ref})$$

$$y' = y$$

ii.  An *y*-direction shear relative to the line $x = x_{ref}$ is generated with transformation matrix given as :
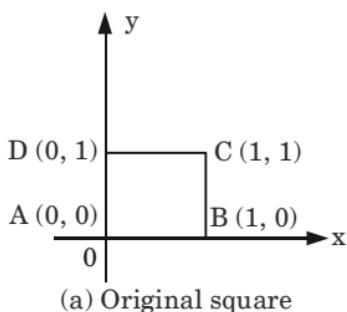
$$Sh_{yref} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & -Sh_y.x_{ref} \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x$$
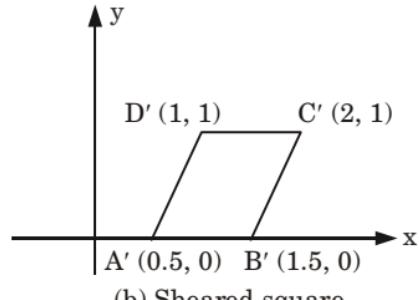
$$y' = Sh_y(x - x_{ref}) + y$$

**Example :** Consider a square with $A(0, 0)$, $B(1, 0)$, $C(1, 1)$ and $D(0,1)$ and apply shearing transformation with shear parameter $Sh_x = 0.5$ and relative to the line $y_{ref} = -1$

$$\therefore \quad \begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ -Sh_x \cdot y_{ref} & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.5 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0 & 1 \\ 1.5 & 0 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



(a) Original square      (b) Sheared square

**Fig. 2.10.3.**

**Que 2.11.** **Prove that simultaneous shearing in both direction ($x$ and $y$ directions) is not equal to the composition of pure shear along the $x$-axis followed by pure shear along the $y$-axis.**

**Answer**

We know that the simultaneous shearing ($x$ and $y$-axis) matrix is

$$\begin{pmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad ...(2.11.1)$$

Shearing matrix in $x$-direction = $\begin{pmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Shearing matrix in $y$-direction = $\begin{pmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Shearing in the $x$-direction and then the $y$-direction =

$$\begin{pmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & sh_x & 0 \\ sh_y & 1 + sh_x sh_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad ...(2.11.2)$$

As equation $(2.11.1) \neq (2.11.2)$, hence, simultaneous shearing in both directions is not equal to the composition of pure shear along $x$-axis followed by pure shear along $y$-axis.

Hence proved.

---

### PART-4

*Windowing and Clipping : Viewing Pipeline, Viewing Transformation.*

---

### CONCEPT OUTLINE

- Windowing referred as selected area of a picture for viewing.
- Clipping is the method to cut a specific object.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 2.12.** **What do you understand by the term "Clipping and Windowing" ?**

**Answer**

**Clipping :**

1. Clipping is the procedure that identifies those portions of a picture that are either inside or outside of a specific region of space.

2. Point, line, area or text can be clipped.

3. The region against which an object is to be clipped is called clipping window.

**Windowing :**

1. Windowing is the process of selecting and viewing the picture with different views.

2. An area on a display device to which a window is mapped is called a viewport.

3. A window and viewports are rectangle in standard position, with the rectangle edges parallel to coordinate axis.

---

**Que 2.13.** **What is window to view point coordinate transformation ? What are the issues related to multiple windowing ?**     **AKTU 2015-16, Marks 10**

**Answer**

**Window to view point coordinate transformation :** Window to viewport mapping or transformation is done in following three steps :

**Step 1 :** The object together with its window is translated until the lower-left corner of the window is at the origin.
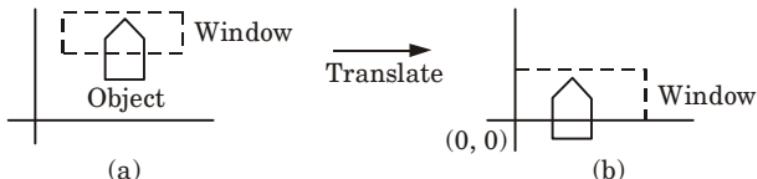


**Fig. 2.13.1.**

**Step 2 :** The object and the window are then scaled until the window has the dimension same as viewport. In other words, we are converting the object into image and window in viewport.
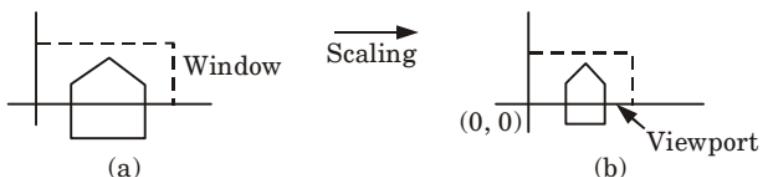


**Fig. 2.13.2.**

**Step 3 :** The final transformation step is another translation to move the viewport to its correct position on the screen.
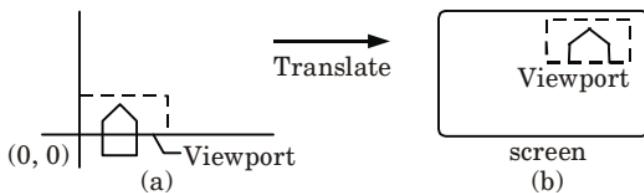
**Fig. 2.13.3.**

**Issues related to multiple windowing :**

1. A key problem in multi-windowing is how to automatically assign windows to the relevant areas within an image.

2. More difficult to work with several applications at once.

### PART-5

*2-D Clipping Algorithm-Line Clipping Algorithm such as Cohen Sutherland Line Clipping Algorithm, Liang-Barsky Algorithm.*

### CONCEPT OUTLINE

- Types of line clipping algorithm are :

  i.  Cohen-Sutherland line clipping

  ii. Liang-Barsky line clipping

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 2.14.** Explain Cohen-Sutherland line clipping algorithm.

**AKTU 2014-15, Marks 06**

**Answer**

**Cohen-Sutherland line clipping algorithm :**

1. In this algorithm, a rectangular window is considered whose coordinates are ($xw_{min}$, $yw_{min}$, $xw_{max}$, $yw_{max}$) as shown in Fig. 2.14.1.
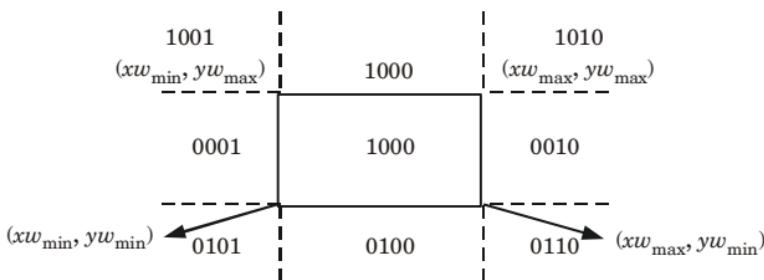
**Fig. 2.14.1.**

2. Every line end point is assigned a 4-digit binary code (region code) that identifies the location of the point corresponding to the boundaries of the clipping window.

3. The procedure for calculating the end point $p(x, y)$ code is as follow :

   **a.  For bit 1**

   If $(y - yw_{max}) < 0$ then

   code = 0;

   else

   code = 1;

   **b.  For bit 2**

   If $(yw_{min} - y) < 0$ then

   code = 0;

   else

   code = 1;

   **c.  For bit 3**

   if$(x - xw_{max}) < 0$ then

   code = 0;

   else

   code = 1;

   **d.  For bit 4**

   if$(xw_{min} - x) < 0$ then

   code = 0;

   else

   code = 1;

4. If the region code of both the end points of a line segment is 0000, the line is completely visible as it completely lies within the window.

5.   If the region code of both the end points of the line segment is not 0000 then we calculate AND operation of both end points $A$ and $B$.

6.   If the result of AND operation is 0000, the line is partially visible and then find the point of intersection.

**Que 2.15.** | **Write Cohen and Sutherland line clipping algorithm. Apply it for calculating the saved portion of a line from (2, 7) to (8, 12) in a window**

$(X_{w\min} = Y_{w\min} = 5$ and $X_{w\max} = Y_{w\max} = 10)$. | **AKTU 2013-14, Marks 06**

**Answer**

**Cohen and Sutherland algorithm :** Refer Q. 2.14, Page 2–20C, Unit-2.

**Numerical :**

Let $P_1 = (2, 7)$ and $P_2 = (8, 12)$, windows lower left corner = (5, 5) windows upper right corner = (10, 10) as shown in Fig. 2.15.1.
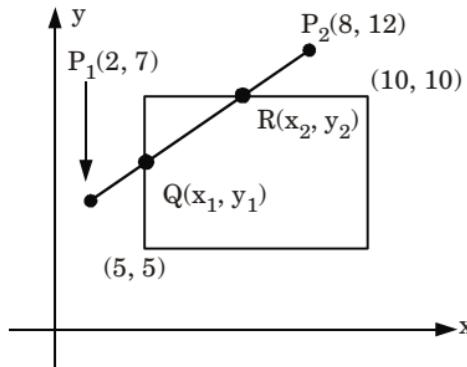


**Fig. 2.15.1.**

Now, we assign the four digit binary code for $P_1$ and $P_2$ shown in Table 2.15.1.

**Table 2.15.1.**

|  |  |  | For $P_1$ (2, 7) | For $P_2$ (8, 12) |
|---|---|---|---|---|
| Bit 1 : | $x - x_{w\min}$ | $x - 5$ | $-3$ | 3 |
| Bit 2 : | $x_{w\max} - x$ | $10 - x$ | 8 | 2 |
| Bit 3 : | $y - y_{w\min}$ | $Y - 5$ | 2 | 7 |
| Bit 4 : | $y_{w\max} - y$ | $10 - y$ | 3 | $-2$ |

If sign bit is negative, it will assign unity(1) and if it is positive, it will assign zero(0).

Binary code for $P_1(2, 7)$ : | bit 4 | bit 3 | bit 2 | bit 1 | 0001

Binary code for $P_2(8, 12)$ : | bit 4 | bit 3 | bit 2 | bit 1 | 1000

So, AND of $P_1$ and $P_2$ = 0000

Line $P_1P_2$ intersects the window at $Q(x_1, y_1)$ and $R(x_2, y_2)$. To find the intersection points we use equation of line $P_1P_2$.

Slope of line $P_1P_2$ is $m = \dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{12 - 7}{8 - 2} = \dfrac{5}{4}$

Equation of line $P_1P_2$ is given as

$$y = mx + c$$

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right) x + c$$

$$y = \frac{5}{4} x + c \qquad \qquad ...(2.15.1)$$

$\because$   $P(2, 7)$ lies on the line so it satisfies the equation (2.15.1)

Put $x = 2$ and $y = 7$ in equation (2.15.1)

$$7 = \frac{5}{4} \times 2 + c$$

$$c = 7 - \frac{5}{2} = \frac{9}{2} = 4.5$$

Equation of line $P_1P_2$ is given as

$$y = \frac{5}{4} x + 4.5 \qquad \qquad ...(2.15.2)$$

**For intersection point $Q(x_1, y_1)$ :**

Since, $Q(x_1, y_1)$ lies on line and the boundary of window. So, it satisfy the equation (2.15.2) and $x_1 = 5$

$$y_1 = \frac{5}{4} x_1 + 4.5$$

$$4y_1 - 5x_1 = 18$$

Put $x_1 = 5$, we get

$$4y_1 = 18 + 5 \times 5$$

$$y_1 = \frac{18 + 25}{4} = \frac{43}{4} = 10.75$$

**For intersection point $R(x_2, y_2)$ :**

Since, $R(x_2, y_2)$ lies on line and the boundary of window. So, $y_2 = 10$

$$y_2 = \frac{5}{4} x_2 + 4.5$$

$$4y_2 - 5x_2 = 18$$

Put $y_2 = 10$

$$40 - 5x_2 = 18$$

$$x_2 = \frac{40 - 18}{5} = 4.4$$

So, the point of intersection are (5, 10.75) and $R(4.4, 10)$

Visible line segment is $Q(5, 10.75)$ to $R(4.4, 10)$

---

**Que 2.16.** Given a clipping window $A(20, 20)$, $B(60, 20)$, $C(60, 40)$, $D(20, 40)$. Using Sutherland Cohen algorithm find the visible portion of line segment joining the points $P(40, 80)$, $Q(120, 30)$.

**Answer**

D(20, 40)                    C(60, 40)

A(20, 20)                    B(60, 20)
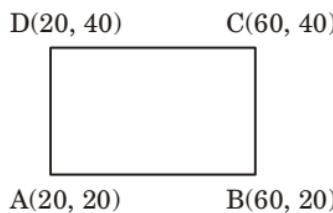
**Fig. 2.16.1.**

Here                    $x_L = 20 \quad y_B = 20$

$x_R = 60 \quad y_T = 40$

The outcodes can be calculated as

Bit 1 = sign of $(y - y_T)$

Bit 2 = sign of $(y_B - y)$

Bit 3 = sign of $(x - x_R)$

Bit 4 = sign of $(x_L - x)$

and sign = 1 if value is +ve and sign = 0 if value is –ve.

Thus, the outcodes of $P(40, 80)$ is 1000 and $Q(120, 30)$ is 0010.

Both endpoint codes are not zero and their logical AND is zero. Hence, line cannot be rejected as invisible.

Slope of $PQ(m) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{30 - 80}{120 - 40} = -\frac{5}{8}$

and intersections point are calculated as :

**Left intersection point :**

$$y = m(x_L - x_1) + y_1 = -\frac{5}{8} \ (20 - 40) + 80$$

= 92.5, which is greater than $y_T$ and hence rejected.

**Right intersection point :**

$$y = m(x_R - x_1) + y_1 = -\frac{5}{8} \ (60 - 40) + 80$$

= 67.5, which is greater than $y_T$ and hence rejected.

**Top intersection point :**

$$x = x_1 + \frac{1}{m} \ (y_T - y_1) = 40 + \frac{1}{-\frac{5}{8}} \ (40 - 80)$$

= 104, which is greater than $x_R$ and hence rejected.

**Bottom intersection point :**

$$x = x_1 + \frac{1}{m} \ (y_B - y_1) = 40 + \frac{1}{-\frac{5}{8}} \ (20 - 80)$$

= 136, which is greater than $x_R$ and hence rejected.

Since both the values of $y$ are greater than $y_T$ and $x_R$. Therefore the line segment *PQ* is completely outside the window.

---

**Que 2.17.** Write Liang and Barsky line clipping algorithm. Apply it for calculating the saved portion of line from (2,7) to (8,12) in a window.

$(x_{w\min} = y_{w\min} = 5$ and $x_{w\max} = y_{w\max} = 10)$. | **AKTU 2012-13, Marks 06**

**Answer**

**The Liang-Barsky algorithm :**

1. Read two endpoint of the line say $P_1 \ (x_1, y_1)$ and $P_2 \ (x_2, y_2)$.

2. Read two corners (left-top and right-bottom) of the window, say $(x_{w\min}, y_{w\max}, x_{w\max}, y_{w\min})$.

3. Calculate the values of parameters $p_i$ and $q_i$ for $i = 1, 2, 3, 4$ such that

$$p_1 = -\Delta x \qquad\qquad q_1 = x_1 - x_{w\min}$$
$$p_2 = \Delta x \qquad\qquad q_2 = x_{w\max} - x_1$$

$$p_3 = -\Delta y \qquad\qquad q_3 = y_1 - y_{w\min}$$

$$p_4 = \Delta y \qquad\qquad q_4 = y_{w\max} - y_1$$

4.   If $p_i = 0$, then

      {   The line is parallel to $i^{\text{th}}$ boundary.

        if $q_i < 0$ then

      {   line is completely outside the boundary, can be eliminated

        and goto stop.

      }

      else

      {

      line is inside the boundary and needs further consideration

      }

      else

      {

      Calculate $r_i = \dfrac{q_1}{p_i}$ for $i = 1, 2, 3, 4$

      }

    }

5.   For all $k$ such that $p_k < 0$ calculate $r_k = q_k/p_k$. Let $u_1$ be the maximum of the set containing 0 and the various values of $r$.

6.   For all $k$ such that $p_k > 0$ calculate $r_k = q_k/p_k$. Let $u_2$ be the maximum of the set containing 1 and the calculated $r$ values.

7.   If $u_1 > u_2$, eliminate the line since it is completely outside the clipping window, otherwise use $u_1$ and $u_2$ to calculate the endpoint of the clipped line.

8.   Stop.

**Numerical :**

$$P_1 = (2, 7)$$
$$P_2 = (8, 12)$$
$$(x_{w\min}, y_{w\min}) = (5, 5)$$
$$(x_{w\min}, y_{w\max}) = (5, 10)$$

$$(x_{w\max}, y_{w\min}) = (10, 5)$$

$$(x_{w\max}, y_{w\min}) = (10, 10)$$

$$\Delta x = 8 - 2 = 6$$

$$\Delta y = 12 - 7 = 5$$

$$p_1 = -\Delta x = -6 \qquad q_1 = x_1 - x_{w\min} = 2 - 5 = -3$$

$$p_2 = \Delta x = 6 \qquad q_2 = x_{w\max} - x_1 = 10 - 2 = 8$$

$$p_3 = -\Delta y = -5 \qquad q_3 = y_1 - y_{w\min} = 7 - 5 = 2$$

$$p_4 = \Delta y = 5 \qquad q_4 = y_{w\max} - y_1 = 10 - 7 = 3$$

$$r_1 = \frac{q_1}{p_1} = \frac{-3}{-6} = \frac{1}{2}$$

$$r_2 = \frac{q_2}{p_2} = \frac{8}{6} = \frac{4}{3}$$

$$r_3 = \frac{q_3}{p_3} = \frac{2}{-5} = -\frac{2}{5}$$

$$r_4 = \frac{q_4}{p_4} = \frac{3}{5} = \frac{3}{5}$$

$$\text{or } (p_i < 0) \; u_1 = \text{Max}\left(\frac{1}{2}, -\frac{2}{5}, 0\right) = \frac{1}{2}$$

$$\text{or } (p_i > 0) \; u_2 = \text{Min}\left(\frac{4}{3}, \frac{3}{5}, 1\right) = \frac{3}{5}$$

$$u_1 < u_2$$

$$\therefore \qquad x'_1 = x_1 + u_1\Delta x = 2 + \frac{1}{2}\,(6) = 5$$

$$x'_2 = x_1 + u_2\Delta x = 2 + \frac{3}{5}\,(6) = 5.6$$

$$y'_1 = y_1 + u_1\Delta y = 7 + \frac{1}{2}\,(5) = 9.5$$

$$y'_2 = y_1 + u_2\Delta y = 7 + \frac{3}{5}\,(5) = 10$$

Coordinates of clipped lines will be (5, 9.5) and (5.6, 10).
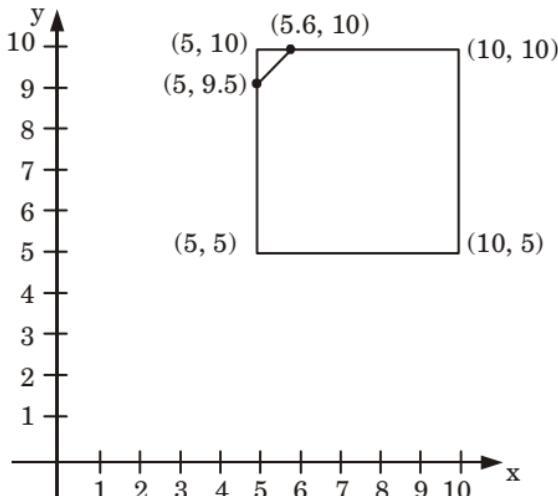
**Fig. 2.17.1.**

**Que 2.18.** | Apply the Liang-Barsky to clip the line segment from *A* (3, 7) to *B* (8, 10) against the regular rectangular window *P* (1, 2), *Q* (9, 2), *R* (9, 8) and *S* (1, 8). | **AKTU 2014-15, Marks 07**
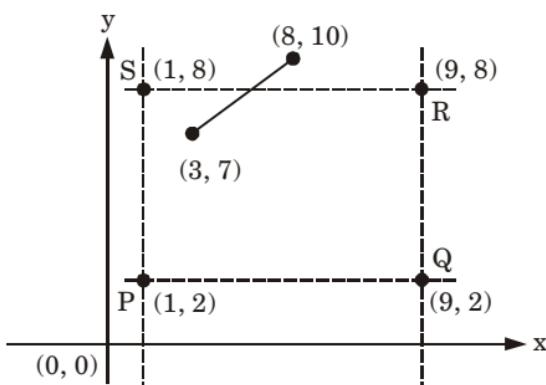
**Answer**



**Fig. 2.18.1.**

Point of line segments are (3, 7) and (8, 10).

Rectangular window points are *P*(1, 2), *Q*(9, 2), *R*(9, 8) and *S*(1, 8).

$$x_{max} = 9 \quad y_{max} = 8$$
$$x_{min} = 1 \quad y_{min} = 2$$

$$(x_1, y_1) = (3, 7)$$
$$(x_2, y_2) = (8, 10)$$
$$\Delta x = x_2 - x_1 = 8 - 3 = 5$$
$$\Delta y = y_2 - y_1 = 10 - 7 = 3$$

| | |
|---|---|
| $p_1 = -\Delta x = -5$ | $q_1 = x_1 - x_{min} = 3 - 1 = 2$ |
| $p_2 = \Delta x = 5$ | $q_2 = x_{max} - x_1 = 9 - 3 = 6$ |
| $p_3 = -\Delta y = -3$ | $q_3 = y_1 - y_{min} = 7 - 2 = 5$ |
| $p_4 = \Delta y = 3$ | $q_4 = y_{max} - y_1 = 8 - 7 = 1$ |

$$r_1 = q_1 / p_1 = \frac{2}{-5} = -0.4$$

$$r_2 = q_2 / p_2 = \frac{6}{5} = 1.2$$

$$r_3 = q_3 / p_3 = \frac{5}{-3} = -1.66$$

$$r_4 = q_4 / p_4 = \frac{1}{3} = 0.33$$

$$u_1 = \max(-0.4, -1.66, 0) = 0$$
$$u_2 = \min(1, 1.2, 0.33) = 0.33$$
$$u_1 < u_2 \text{ endpoints are visible}$$
$$x'_1 = x_1 + (\Delta x \times u_1) = 3 + (5 \times 0) = 3$$
$$y'_1 = y_1 + (\Delta y \times u_1) = 7 + (3 \times 0) = 7$$
$$x'_2 = x_1 + (\Delta x \times u_2) = 3 + (5 \times 0.33) = 4.65$$
$$y'_2 = y_1 + (\Delta y \times u_2) = 7 + (3 \times 0.33) = 7.99$$

∴   Visible line will be $P_1(3, 7)$ and $P_2(4.65, 7.99)$.

**Que 2.19.** **Write an algorithm for Cohen Sutherland line clipping algorithm. Compare it with Liang-Barsky line clipping algorithm.**

AKTU 2015-16, Marks 10

**Answer**

**Cohen Sutherland algorithm :** Refer Q. 2.14, Page 2–20C, Unit-2.

**Comparison :**

| S. No. | Cohen Sutherland line clipping algorithm | Liang-Barsky line clipping algorithm |
|--------|------------------------------------------|--------------------------------------|
| 1. | It is not the most efficient but efficient when most of the lines to be clipped are either rejected or accepted. | It is more efficient than Cohen Sutherland since intersection calculations are reduced. |
| 2. | Each intersection calculation requires both multiplication and a division. | Each update of parameters requires only one division. |
| 3. | It repeatedly calculates intersection along a line path even through the line may be completely outside the clip window. | Window intersections of lines are calculated once when final values have been computed. |

---

### PART-6

*Line Clipping against Non-Rectangular Clip Window, Polygon Clipping-Sutherland Hodgeman Polygon Clipping, Weiler and Atherton Polygon Clipping.*

---

### CONCEPT OUTLINE

- A polygon is represented as a number of line segments connected end to end to form a closed figure.
- Weiler Atherton clipping algorithm allows clipping of a subject polygon by an arbitrarily shaped clip polygon.
- Sutherland-Hodgeman polygon clipping algorithm clips convex polygon.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 2.20.** **What do you understand by polygon ? Define convex and concave polygon.**

**Answer**

**Polygon :**

1. A polygon may be represented as a number of line segments connected end to end to form a closed figure.

2. It may be represented as the points where minimum three vertices and three edges are connected to form a closed figure.

3. Polygons can be divided into following two classes :

    **i.** **Convex polygon :** A polygon is called convex if the line joining any two interior points of the polygon lie completely inside the polygon.
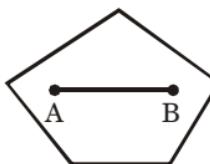


**Fig. 2.20.1.**

    **ii.** **Concave polygon :** A non-convex polygon is said to be concave.
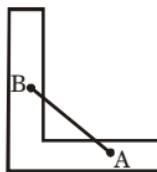


**Fig. 1.20.2.**

**Que 2.21.** **How inside test are performed using odd-parity rules or winding number methods ?**

**Answer**

1. Inside-outside test states, "How can we determine whether or not a point is inside of a polygon".

2. Following are the two methods to identify this :

    **a.** **Odd-parity methods :**

      i. Following is a simple idea to check whether a point is inside or outside.

        1. Draw a horizontal line to the right of each point and extend it to infinity.

        2. Count the number of times the line intersects with polygon edges.

3. A point is inside the polygon if either count of intersections is odd or point lies on an edge of polygon. If none of the conditions is true, then point lies outside.



count = 1     odd
count = 2     even

count = 3     odd
count = 4     even
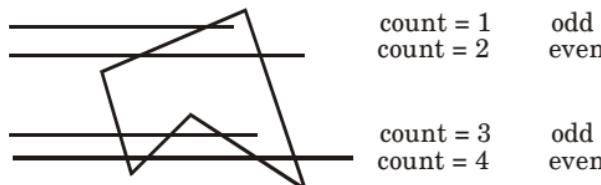
**Fig. 2.21.1.**

ii. If the point of intersection is also the vertex where two sides met then to handle this case we must look at the other endpoints of the two segments which meet at this vertex.

iii. If these points lie on the same side of the constructed line, then the point in question counts as an even number of intersection.

iv. If they lie opposite side of the constructed line, then the point is counted as a single intersection.



Counts odd

Counts even
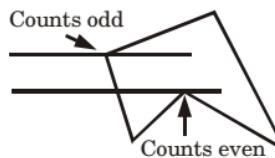
**Fig. 2.21.2.**

**b.  Winding number method :**

i. In this method every point has a winding number, and the interior points of a two dimensional object are defined to be those that have a nonzero value for the winding number :

1. Initializing the winding number to 0.

2. Imagine a line drawn from any position $P$ to a distant point beyond the coordinate extents of the object.

3. Count the number of edges that cross the line in each direction.

4. We add 1 to the winding number every time we intersect a polygon edge that crosses the line from right to left, and we subtract 1 every time we intersect an edge that crosses from left to right.

5. If the winding number is non-zero then point $P$ is inside otherwise $P$ is outside the polygon.
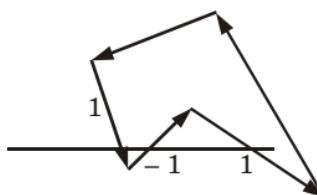
**Fig. 2.21.3.**

Que 2.22. | What are the steps required to fill a polygon using flood-fill or boundary fill ?

**OR**

**Write the algorithm for filling polygons and explain it with a suitable example.**

**Answer**

Filling is the process of 'coloring in' a fixed area or region. Following are the various algorithm used to fill the area :

**1.    Boundary-fill algorithm :**

a.    An approach to area filling is to start at a point inside a region and point the interior outward toward boundary.

b.    If the boundary is specified in a single color, the fill algorithm proceeds outward pixel by pixel until the boundary color is encountered.

c.    It is particularly useful in interactive painting packages, where interior points are easily selected.

d.    A boundary fill procedure accepts as input the coordinates of an interior points $(x, y)$ a fill color, and a boundary color.

e.    The following procedure illustrate a recursive method for filling 4-connected with an intensity specified in parameter fill up to boundary specified with parameter boundary.



4-connected        8-connected
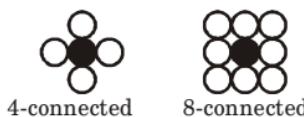
**Fig. 2.22.1.**

Procedure boundaryfill4 $(x, y$, fill, boundary : integer);

var

current : integer;

begin

current = getpixel (*x*, *y*);

if (current != boundary) and (current != fill) then

begin

setpixel(*x*, *y*, fill);

boundaryfill4 (*x* + 1, *y*, fill, boundary);

boundaryfill4 (*x* – 1, *y*, fill, boundary);

boundaryfill4 (*x*, *y* + 1, fill, boundary);

boundaryfill4 (*x*, *y* – 1, fill boundary);

end;

**2.  Flood-fill algorithm :**

a.  It is used to fill the area that is not defined within a single color boundary.

b.  We can paint such areas by replacing a specified interior color.

c.  We start from a specified interior point (*x*, *y*) and reassign all pixels values that are currently set to a given interior color with the desired fill color.

d.  The following procedure flood fills a 4-connected region recursively, starting from the input position.

Procedure floodfill4(*x*, *y*, fillcolor, oldcolor : integer);

begin

if (getpixel(*x*, *y*) = oldcolor) then

begin

setpixel(*x*, *y*, fillcolor);

floodfill4 (*x*+1, *y*, fillcolor, oldcolor);

floodfill4 (*x*–1, *y*, fillcolor, oldcolor);

floodfill4 (*x*, *y*+1, fillcolor, oldcolor);

floodfill4 (*x*, *y*–1, fillcolor, oldcolor);

end;

end;

**Que 2.23.** | **Discuss Weiler-Atherton algorithm.**

**Answer**

1. The Weiler-Atherton is an algorithm which is used where clipping of polygon is needed.

2. It allows clipping of a subject or candidate polygon by an arbitrarily shaped clipping polygon/area/region.

3. It is generally applicable only in 2D. However, it can be used in 3D through visible surface determination and with improved efficiency through *z*-ordering.

4. Consider a polygon $A$ as the clipping region and polygon $B$ as the subject polygon to be clipped, the algorithm consists of the following steps :

   a. List the vertices of the clipping-region polygon $A$ and those of the subject polygon $B$.

   b. Label the listed vertices of subject polygon $B$ as either inside or outside of clipping region $A$.

   c. Find all the polygon intersections and insert them into both lists, linking the lists at the intersections.

   d. Generate a list of inbound intersections *i.e.*, the intersections where the vector from the intersection to the subsequent vertex of subject polygon $B$ begins inside the clipping region.

   e. At occurrence of leaving intersection the algorithm follows the clip polygon vertex list from leaving vertex in downward direction.

   f. At occurrence of entering intersection the algorithm follows subject polygon vertex list from the entering intersection vertex. This process is repeated till we get the starting vertex.

   g. This process has to be repeated for all remaining entering intersections which are not included in the previous traversing of vertex list.

5. This algorithm is used for clipping concave polygons. Here $V_1$, $V_2$, $V_3$, $V_4$, $V_5$ are the vertices of the polygon. $C_1$, $C_2$, $C_3$, $C_4$ are the vertices of the clip polygon and $I_1, I_2, I_3, I_4$ are the intersection points of polygon and clip polygon.

6. In this algorithm we take a starting vertex like $I_1$ and traverse the polygon like $I_1, V_3, I_2$.

7. Since $I_3$ is not included in first traverse, hence, we start the second traversal from $I_3$.

Therefore, first traversal gives polygon as : $I_1$, $V_3$, $I_2$, $I_1$ and second traversal gives polygon as : $I_3$, $V_5$, $I_4$, $I_3$.
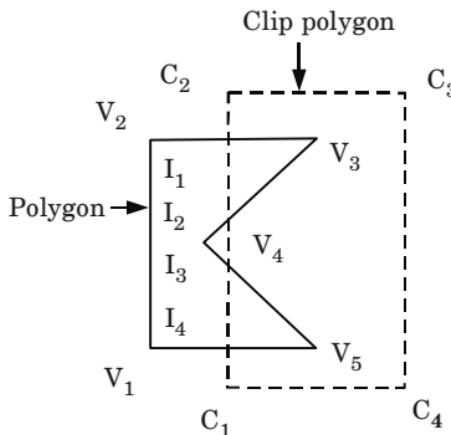
**Fig. 2.23.1.**

Que 2.24. | Write the Sutherland-Hodgeman polygon clipping algorithm. Explain the modification given by Weiler and Atherton for a concave polygon. | AKTU 2012-13, 2013-14; Marks 10

**Answer**

**Sutherland-Hodgeman polygon clipping algorithm :**

1. Read coordinates of all vertices of the polygon.

2. Read coordinates of the clipping window.

3. Consider the left edge of the window.

4. Compare the vertices of each edge of the polygon, individually with the clipping plane.

5. Save the resulting intersections and vertices in the new list of vertices according to four possible cases between the edge and the clipping boundary.

6. Repeat the step 4 and 5 for remaining edges of the clipping window. Each time the resultant list of vertices is successively passed, move the process to the next edge of the clipping window.

7. Stop.

**Weiler-Atherton polygon clipping :**

1. In this algorithm, the vertex processing procedure is modified so that the concave polygons are displayed correctly.

2. This algorithm depends on identifying surfaces as shown in Fig. 2.24.1.

3. There are two directions (clockwise or anticlockwise) that exist to process the polygon vertices.

4.    For clockwise processing of polygon vertices, we use the following rules :

   a.    For an outside to inside pair of vertices, follow the polygon boundary.

   b.    For an inside to outside pair of vertices, follow the window boundary in a clockwise direction.



Clipping a concave polygon by                    Output (two separate
applying Weiler-Atherton algorithm                polygon areas)
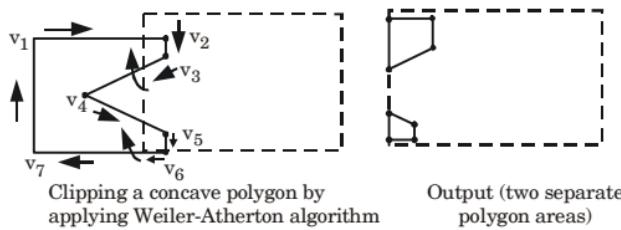
**Fig. 2.24.1.**

PART-7

*Curve Clipping, Text Clipping.*

CONCEPT OUTLINE

- In curve clipping, we clip a curved object against a general polygon clip region.

- Text clipping is a clipping in which whole character or only part of it is clipped.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

Que 2.25.   **Write a short note on curve clipping and text clipping.**

**Answer**

**Curve clipping :**

i.    Curve clipping procedures will involve non-linear equations, however, this requires more processing than for objects with linear boundaries.

ii.   The bounding rectangle for a circle or other curved object can be used first to test for overlap with a rectangular clip window.
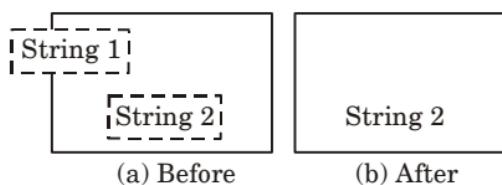
iii.  If the bounding rectangle for the object is completely inside the window, we save the object.

iv.  If the bounding rectangle is determined to be completely outside the window, we discard the object.

v.  This procedures can be applied when clipping a curved object against a general polygon clip region. On the first pass, we can clip the bounding rectangle of the object against the bounding rectangle of the clip region.

vi.  If the two regions overlap, we will need to solve the simultaneous line-curve equations to obtain the clipping intersection points.

**Text clipping :**

i.  Text clipping is a clipping in which we clip the whole character or only part of it and depends on the requirement of the application.

ii.  Following are the various text clipping methods :

   **a.  All-or-none string clipping method :**

   1.  In this method, if all of the string is inside a clip window, we keep it. Otherwise, the string is discarded.

   2.  This method is implemented by considering a bounding rectangle around the text pattern.

   3.  The boundary positions of the rectangle are then compared to the window boundaries, and the string is rejected if there is any overlap.

   4.  This method produces the fastest text clipping.
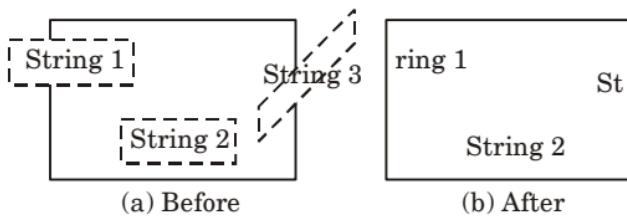


**Fig. 2.25.1.** All-or-none string clipping method.

   **b.  All-or-none character clipping method :**

   1.  In this method, we keep the character of the strings which lies inside clip window, otherwise, discard it.

   2.  In this case, the boundary limits of individual characters are compared to the window.

3. Any character that either overlaps or is outside a window boundary is clipped.



**Fig. 2.25.2.** All-or-none character clipping method.

☺☺☺

# 3

**UNIT**

# Three Dimensional

# CONTENTS

PART- 1

*Three Dimensional : 3D Geometric Primitive, 3D Object Representation.*

## CONCEPT OUTLINE

- Geometric primitives are the basic geometrical shapes used to construct graphics scenes and the resulting final images.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

Que 3.1. | **What do you mean by geometric primitives ? Discuss.**

**Answer**

Geometric primitives are the basic geometrical shapes used to construct computer graphics scenes and the resulting final images.

Commonly used 3D geometric primitives include :

1. **Points :** A point is an exact position or location on a plane surface.

2. **Line :** A line is a straight curve. When geometry is used to model the real world, lines are used to represent straight objects with negligible width and height.

3. **Line segments :** A line segment is a part of a line that is bounded by two end points, and contains every point on the line between its end points.

4. **Planes :** A plane is the two dimensional analogue of a point (zero-dimensions), a line (one-dimension) and a space (three-dimensions).

5. **Circles :** A circle is a simple shape of Euclidean geometry consisting of those points in a plane which is equidistant from a given point called the centre.

6. **Ellipses :** Ellipses are closed curves and are the bounded case of the conic sections, the curves that result from the intersection of a circular cone and a plane that does not pass through its apex.

7. **Triangles :** A polygon with three corners or vertices and three sides or edges which are line segments is called a triangle.

8. **Polygons :** A polygon is traditionally a plane figure that is bounded by a closed path or circuit, composed of a finite sequence of straight line segments.

**9. Spline :** A spline is a special function defined piecewise by polynomials and preferred to polynomial interpolation.

**10. Spheres :** A sphere is a perfectly round geometrical object in three dimensional space, such as the shape of a round ball.

---

**Que 3.2.** | **Discuss 3D representation of objects.**

**Answer**

To represent a 3D object we have categorised the object into some model representation :

1. Polygon and quadric surfaces provide precise descriptions for simple Euclidean objects such as polyhedrons and ellipsoids.

2. Spline surfaces end construction techniques are useful for designing aircraft wings, gears, and other engineering structures with curved surfaces.

3. Procedural methods, such as fractal constructions and particle systems, allow us to give accurate representations for clouds, clumps of grass, and other natural objects.

4. Physically based modeling methods using systems of interacting forces can be used to describe the non-rigid behavior of a piece of cloth.

5. Visualization techniques are applied to three dimensional discrete data sets to obtain visual representations of the data.

6. Representation schemes for solid objects are often divided into two broad categories :

   a. Boundary representations (B-reps) describe a three dimensional object as a set of surfaces that separate the object interior from the environment.

   a. Space partitioning representations are used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids (usually cubes).

---

### PART-2

*3D Transformation, 3D Viewing.*

---

### CONCEPT OUTLINE

- In 3D transformation we can change the shape, position and direction of any object with respect to any coordinates system.
- A 3D viewing in general referred as projection.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 3.3.** **What is 3D transformation ? Write the matrix of each transformation.**

**OR**

**Write short note on 3D transformation.**

**Answer**

**3D transformation :**

1. In a three dimensional homogeneous coordinate representation, a point is translated from position $P(x, y, z)$ to position $P'(x', y', z')$ with matrix operation
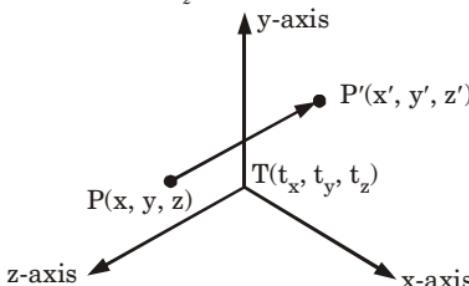
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = T.P$$
$$x' = x + t_x$$
$$y' = y + t_y$$
$$z' = z + t_z$$



**Fig. 3.3.1.**

2. An object is translated in three dimensions by transforming each of the defining points of the object.

**Rotation :**

1. In 3D rotation transformation a point is rotated from $P(x, y, z)$ to $P'(x', y', z')$ and require an angle of rotation and an axis of rotation.

2. Matrix representation of rotation about $z$-axis is given as :

$$x' = x \cos \theta - y \sin \theta$$
$$y' = x \sin \theta + y \cos \theta$$
$$z' = z$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Scaling :** The matrix expression for the scaling transformation of a position $P(x, y, z)$ relative to the coordinate origin can be written as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = S.P$$

where
$$x' = x.s_x$$
$$y' = y.s_y$$
$$z' = z.s_z$$

**Que 3.4.** Derive rotation about $x$-axis, $y$-axis and $z$-axis matrices in 3D. Also, prove that for any rotation matrix $R^{-1}(\theta) = R(-\theta)$.

| AKTU 2012-13, Marks 10 |

| AKTU 2013-14, Marks 06 |

**Answer**

**i. Rotation about $x$-axis (*i.e.* axis of rotation is $x$) :**
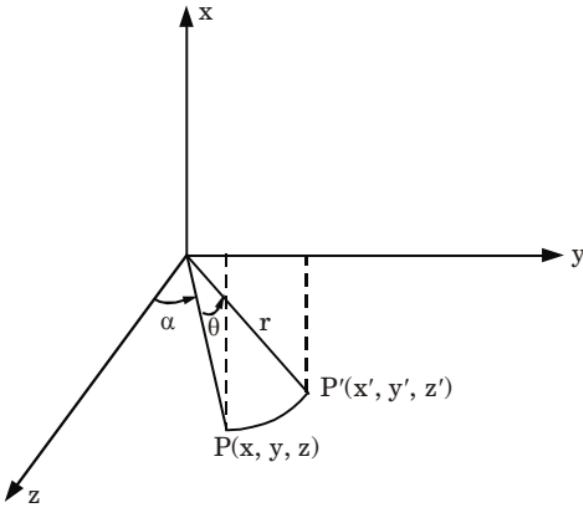


**Fig. 3.4.1.** Rotation when axis of rotation is $x$-axis.

$$x' = x$$
$$y' = r \cos (\alpha + \theta)$$
$$y' = r \cos \alpha \cos \theta - r \sin \alpha \sin \theta \qquad ...(3.4.1)$$

Since $r \cos \alpha = y$ and $r \sin \alpha = z$

Put the value in equation (3.4.1) we get,

$$y' = y \cos \theta - z \sin \theta$$
$$z' = r \sin (\alpha + \theta)$$
$$= r \sin \alpha \cos \theta + r \cos \alpha \sin \theta$$
$$= z \cos \theta + y \sin \theta$$

or,                 $R_{\theta, i} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$

The points after rotation can be calculated by the following equation :

$$P'(x', y', z') = R_{\theta, i} P(x, y, z)$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**ii.  Rotation about *y*-axis (*i.e.*, *y*-axis is axis of rotation) :**



**Fig. 3.4.2.**

$$x' = x \cos\theta + z \sin\theta$$
$$y' = y$$
$$z' = -x \sin\theta + z \cos\theta$$
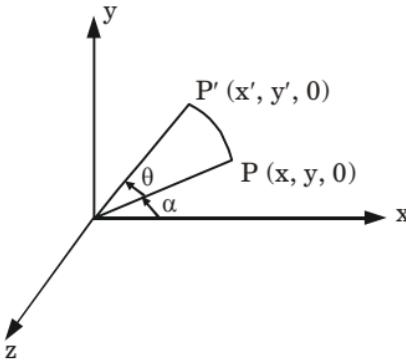
or,                 $R_{\theta, i} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

### iii. Rotation about the *z*-axis (*i.e.*, *z*-axis is axis of rotation)



**Fig. 3.4.3.**

$$x' = x \cos \theta - y \sin \theta$$
$$y' = z \cos \theta + y \sin \theta$$
$$z' = z$$

or,

$$R_{\theta, i} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Numerical :**

Let

$$R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Put $\theta = (-\theta)$, we get

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) & 0 \\ 0 & \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since $\cos(-\theta) = \cos(\theta)$, $\sin(-\theta) = -\sin \theta$, we have

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now,

$$R(-\theta) * R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

We have,

$$R(-\theta) . R(\theta) = I$$

Multiply $R^{-1}(\theta)$ both sides,

$$R(-\theta).R(\theta).R^{-1}(\theta) = I.R^{-1}(\theta)$$

$$R(-\theta) = R^{-1}(\theta)$$

Hence proved.

**Que 3.5.** **Write the steps involved in constructing 3D viewing transformation.**

**Answer**

**The steps involved in constructing 3D viewing transformations :**
The complete three dimensional viewing process (without hidden surface removal) is described by the following steps :

1. Transform coordinates from world coordinates to normalized viewing coordinates by applying the transformation $N_{\text{par}}$ or $N_{\text{per}}$.

2. Clip in normalized viewing coordinates against the canonical clipping volumes.

3. Project onto the screen projection plane using the projections *Par* or *Per*.

4. Apply the appropriate (two dimensional) viewing transformation.

   In terms of transformations, we can describe the above process in terms of a viewing transformation $V_T$, where

   $$V_T = V_2 \, Par \, CL \, N_{\text{par}} \text{ or } V_T = V_2 \, Per \, CL \, N_{\text{per}}$$

   Here $CL$ and $V_2$ refer to the appropriate clipping operations and two dimensional viewing transformations.

---

**PART-3**

*Projection.*

## CONCEPT OUTLINE

- Projection is a process of representing a three dimensional object or scene into two dimensional medium *i.e.*, projection is a shadow of the object.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 3.6.** **What do you mean by projection ? Explain the types of projection differentiate between parallel and perspective projection.**

**Answer**

**Projection :**

1. Projection is a process of representing a three dimensional object or scene into two dimensional medium *i.e.*, projection is a shadow of the object.

2. Projections transform points in a coordinate system of dimension '$n$' into points in a coordinate system of dimension less than $n$.

**Types of projections :**

**1. Parallel projection :**

   a. In a parallel projection, coordinate positions are transformed to the view plane along parallel lines.

   b. These are linear transforms that are useful in blueprints to produce scale drawings of three dimensional objects.

**2. Perspective projection :**

   a. In a perspective projection, object positions are transformed to the view plane along lines that converge to a point called the center of projection.

   b. The projected view of an object is determined by calculating the intersection of the projection lines with the view plane.

**Difference between parallel and perspective projection :**

| S. No. | Parallel projection | Perspective projection |
|--------|---------------------|------------------------|
| 1. | In parallel projection, parallel lines from each vertex on the object are extended until they intersect the view plane. | Perspective projection produces realistic view. |
| 2. | Lines of projection are parallel. | Lines of projection are not parallel. |
| 3. | These are linear transform. | These are non-linear transform. |
| 4. | In parallel projection, the center of projection is at infinity. | In perspective projections, the center of projection is at a point. |

---

**Que 3.7.** **What do you understand by parallel projection ? Also describe orthographic and oblique projection.**

**Answer**

**Parallel projection :** Refer Q. 3.6, Page 3–9C, Unit-3.

**Orthographic projection :**

1. It is a part of the parallel projection in which the center of projection lies at infinity.

2. In orthographic projection, the projectors are perpendicular to the projection plane and hence true size and shape of a single plane face of object is obtained.

3. Orthographic projections do not change the length of line segments which are parallel to projection plane and other lines are projected with reduced length.

4. It is the projection on one of the coordinate planes $i.e.$,

$$x = 0, \quad y = 0 \quad \text{or} \quad z = 0$$

5. The matrix for projection onto the $x = 0$, $y = 0$ and $z = 0$ plane is

$$P_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } P_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Oblique projection :**

1. An oblique projection is obtained by projecting points along parallel lines that are not perpendicular to the projection plane.

2. In this projection angle between the projectors and the plane of projection is not equal to 90° *i.e.*, projection are non-perpendicular to view plane.

3. This type of projection is seen in form of shadow of any object due to sunlight.

4. Transformation matrix for oblique projection is given as

$$M = \begin{bmatrix} 1 & 0 & L_1 \cos\theta & 0 \\ 0 & 1 & L_1 \sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where, $L_1$ is the length of line.

**Que 3.8.** Derive perspective projection transformation matrix. Also define anomalies of perspective projection.

**Answer**

**Transformation matrix of perspective projection :**

1. A perspective transformation is determined by prescribing a center of projection and a view plane.

2. The view plane is determined by its view reference point $R_0$ and view plane normal $N$.

3. The object point $p$ is located in world coordinates at $(x, y, z)$. The problem is to determine the image point coordinates $p'(x', y', z')$.

4. The standard perspective projection is shown in Fig. 3.8.1. Here the view plane is the $xy$-plane and center of projection is taken as the point $C(0, 0, -d)$ on negative $z$-axis.

$$x' = \frac{d.x}{z+d}, \, y' = \frac{d.y}{z+d}, \, z' = 0$$

5. The perspective projection transformation between object and image point is non-linear and so cannot be represented as $3 \times 3$ matrix transformation.

6. If we use homogeneous coordinate, the perspective transformation can be represented as $4 \times 4$ matrix :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} d.x \\ d.y \\ 0 \\ z+d \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
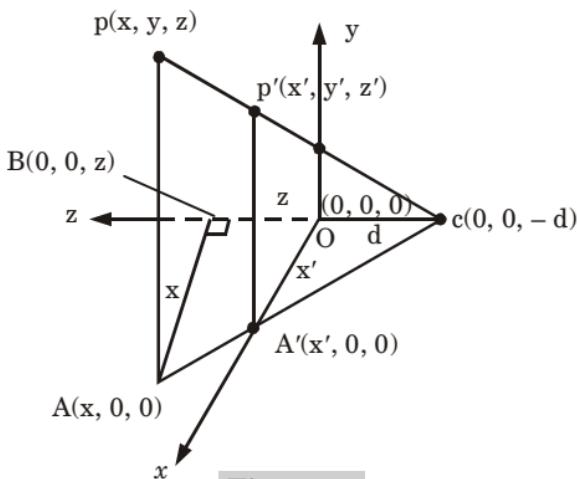
**Fig. 3.8.1.**

**Perspective projection anomalies :**

1. **Perspective foreshortening :** The farther an object is from the center of projection, the smaller it appears.
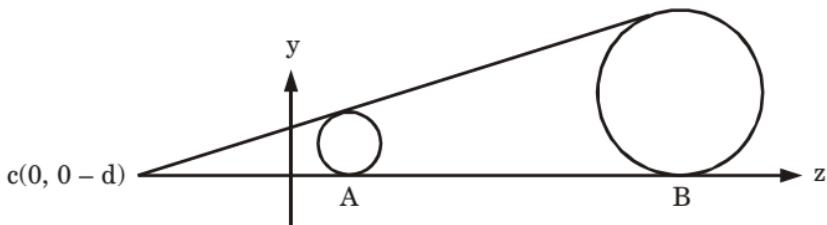


**Fig. 3.8.2.**

2. **Vanishing point :** Projections of lines that are not parallel to the view plane appear to meet at some point on the view plane.

3. **View confusion :** Objects behind the center of projection are projected upside down and backward onto the view plane.

**Que 3.9.** | Derive oblique parallel projection and perspective projection matrices.                    | AKTU 2012-13, 2013-14; Marks 06

**Answer**

**Derivation of oblique parallel projection :**

1. An oblique projection is obtained by projecting points along parallel lines that are not perpendicular to the projection plane.

2. An oblique projection of a point $P(x, y, z)$ on the $z = 0$ plane is shown in Fig. 3.9.1.

3. $P_{ob}$ $(x_p, y_p)$ is oblique projection of the point $P$ and $P_{or}$ $(x, y)$ is the orthographic projection of $P$ on the $z = 0$ plane.

4. Let $L$ be the length of the line joining points $P_{ob}$ $(x_p, y_p)$ and $P_{or}$ $(x, y)$, $\theta$ be the angle made by the line $L$ with $x$-axis.
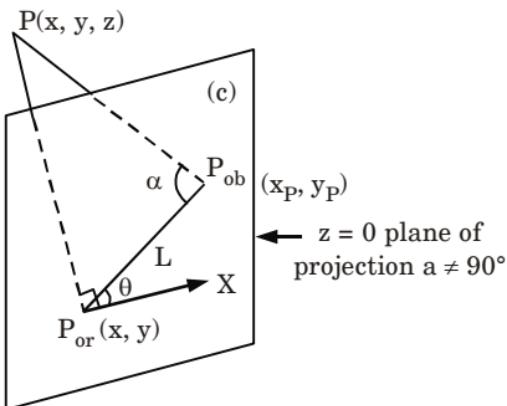
**Fig. 3.9.1.**

5.  We can express the projection coordinates in terms of $x, y, L$ and $\theta$ as
$$x_p = x + L \cos \theta$$
$$y_p = y + L \sin \theta$$

6.  Length $L$ depends on the angle $\alpha$ and the $Z$ coordinate of the point to be projected

$$\tan \alpha = \frac{Z}{L} \Rightarrow L = \frac{Z}{\tan \alpha} = ZL_1 \quad \left( \text{where } L_1 = \frac{1}{\tan \alpha} \right)$$

when $\qquad Z = 1, L = L_1$

7.  Now, we can write the oblique projection as
$$x_p = x + Z(L_1 \cos \theta)$$
$$y_p = x + Z(L_1 \sin \theta)$$

8.  Considering $Z_p = 0$, we get
$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L_1 \cos \theta & 0 \\ 0 & 1 & L_1 \sin \theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Derivation of perspective projection matrix :**

1.  Let us consider the center of projection is at $(x_c, y_c, z_c)$ and the point on object is $(x_1, y_1, z_1)$, then the parametric equation for the line containing these points can be given as

$$x_2 = x_c + (x_1 - x_c)u$$
$$y_2 = y_c + (y_1 - y_c)u$$
$$z_2 = z_c + (z_1 - z_c)u$$

where $u$ is the parameter

2.  For projected point $z_2$ is 0, therefore, the third equation can be written as

$$0 = z_c + (z_1 - z_c)u$$

$$\therefore \qquad u = -\frac{z_c}{(z_1 - z_c)}$$

3.  Put this value of $u$ in first two equations, we get

$$x_2 = x_c - \frac{z_c}{(z_1 - z_c)} (x_1 - x_c)$$

$$= \frac{x_c z_1 - x_c z_c - x_1 z_c + x_c z_c}{(z_1 - z_c)}$$

$$\therefore \qquad x_2 = \frac{x_c z_1 - x_1 z_c}{(z_1 - z_c)}$$

and

$$y_2 = y_c - \frac{z_c}{(z_1 - z_c)} (y_1 - y_c)$$

$$= \frac{y_c z_1 - y_c z_c - y_1 z_c + y_c z_c}{(z_1 - z_c)}$$

$$y_2 = \frac{y_c z_1 - y_1 z_c}{(z_1 - z_c)}$$

4.  So, the homogeneous matrix can be written as

$$[x_2 \quad y_2 \quad z_2 \quad 1] = [x_1 \quad y_1 \quad z_1 \quad 1] \begin{bmatrix} -z_c & 0 & 0 & 0 \\ 0 & -z_c & 0 & 0 \\ x_c & y_c & 0 & 1 \\ 0 & 0 & 0 & -z_c \end{bmatrix}$$

**Que 3.10.** **Explain parallel and perspective projection. Justify the depth cueing projection for 3D display methods.**

**Answer**

**Parallel projection :** Refer Q. 3.6, Page 3–9C, Unit-3.

**Types of parallel projection :**

i.   **Orthographic projection :** Refer Q. 3.7, Page 3–10C, Unit-3.

ii.  **Oblique projection :** Refer Q. 3.7, Page 3–10C, Unit-3.

**Perspective projection :** Refer Q. 3.6, Page 3–9C, Unit-3.

**Different types of perspective projection are :**

1.  **Single-point perspective transformation :** One-point perspective projection occurs when only one principal axis intersects the plane of projection.

2.  **Two-point perspective transformation :** Two-point or two principal vanishing point perspective projection occurs when the plane of projection intersects exactly two of the principal axis.

**3.    Three-point perspective transformation :** Three-point projection is obtained when the projection plane intersects all three of the principal axis *i.e.*, none of the principal axis is parallel to the projection plane.

**Depth cueing projection :**

1.    In depth cueing, we include the depth information in the displayed object.

2.    The depth of an object can be represented by the intensity of the image.

3.    Depth cueing is applied by choosing maximum and minimum intensity values.

4.    The parts of the objects closest to the viewing position are displayed with the highest intensities and objects farther away are displayed with decreasing intensities.

5.    Reason to use depth cueing projection for 3D display method:

   a.    To create realistic view of an object.

   b.    To easily identify the particular viewing direction of displayed object.

6.    A simple method for indicating depth is using wireframe display.

**Que 3.11.**    **Explain the matrix for perspective projection for three vanishing points.**

**Answer**

**Matrix for perspective projection for three vanishing point :**

1.    Three vanishing point projection occurs when the projection plane intersects all three of the principal axis *i.e.*, none of the principal axis is parallel to the projection plane.

2.    The need of three vanishing point perspective transformation is to reconstruct the shape of a 3D object.

3.    The matrix representation of three vanishing point perspective transformation is

$$\begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[x' \ \ y' \ \ z' \ \ 1] = [x \ \ y \ \ z \ \ 1] \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [x \ \ y \ \ z \ \ (px + qy + rz + 1)]$$

$$= \left[ \frac{x}{(px + qy + rz + 1)} \ \frac{y}{(px + qy + rz + 1)} \ \frac{z}{(px + qy + rz + 1)} \ \ 1 \right]$$

4.   It has three center of projections, one on $x$-axis at $[-1/p, 0, 0, 1]$, second on $y$-axis at $[0, -1/q, 0, 1]$ and third on $z$-axis at $[0, 0, -1/r, 1]$.

5.   It also has three vanishing points, one on $x$-axis $[1/p, 0, 0, 1]$, one on $y$-axis at $[0, 1/q, 0, 1]$ and third on $z$-axis at $[0, 0, 1/r, 1]$.

**Que 3.12.** **Find the perspective projection onto the view plane $z = d$ where the centre of projection is the origin $(0, 0, 0)$.**

**Answer**

The plane $z = d$ is parallel to the $xy$ plane. Thus the view plane normal vector $N$ is the same as the normal vector $K$ to the $xy$ plane, $i.e.$, $N = K$. Select view reference point as $R_0(0, 0, d)$, then

$$N(n_1, n_2, n_3) = (0, 0, 1)$$
$$R_0(x_0, y_0, z_0) = (0, 0, d)$$
$$d_0 = n_1x_0 + n_2y_0 + n_3z_0 = d$$

then projection matrix is

$$\text{Per}_{k, R_0} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

**Que 3.13.** **Find the general form of an oblique projection onto the $xy$ plane.**

**Answer**

1.   Oblique projection can be specified by a number $f$ and an angle $\theta$. The number $f$ prescribes the ratio that any line $L$ perpendicular to the $xy$ plane will be foreshortened after projection.

2.   The angle $\theta$ is the angle that the projection of any line perpendicular to the $xy$ plane makes with the $x$-axis.
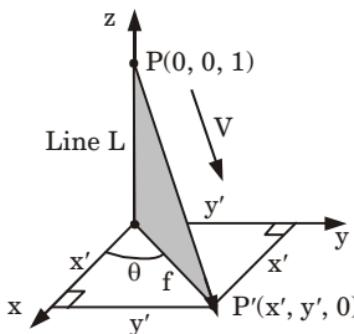


**Fig. 3.13.1.**

3. To determine the projection transformation, we need to find the direction vector $V$. From above figure we have line $L$ of length 1, we see that vector $\overline{P'P}$ has the same direction on $V$.

$$V = \overline{P'P} = x'\hat{i} + y'\hat{j} - \hat{k} = a\hat{i} + b\hat{j} + c\hat{k}$$
$$a = x' = f\cos\theta, \, b = y' = f\sin\theta$$
$$c = -1$$

$$\text{Par}_V = \begin{bmatrix} 1 & 0 & f\cos\theta & 0 \\ 0 & 1 & f\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Que 3.14.** Find the transformation for (*a*) cavalier with $\theta = 45°$ and (*b*) cabinet projection with $\theta = 30°$.

**Answer**

a. A cavalier projection is an oblique projection where there is no foreshortening of line perpendicular to the $xy$ plane. We then see for $f = 1$, with $\theta = 45°$, we have

$$\text{Par}_{V1} = \begin{bmatrix} 1 & 0 & \dfrac{\sqrt{2}}{2} & 0 \\ 0 & 1 & \dfrac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b. A cabinet projection is an oblique projection with $f = \dfrac{1}{2}$, with $\theta = 30°$, we have

$$\text{Par}_{V2} = \begin{bmatrix} 1 & 0 & \dfrac{\sqrt{3}}{4} & 0 \\ 0 & 1 & \dfrac{1}{4} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## PART-4

*3D Clipping*

### CONCEPT OUTLINE

- Clipping in three dimensions can be accomplished using extensions of two dimensional clipping methods.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 3.15.** **What do you mean by 3D clipping ? Discuss in detail.**

**OR**

**Write a short note on 3D clipping.**

**Answer**

1. Clipping in three dimensions can be accomplished using extensions of two dimensional clipping methods.

2. In 3D clipping we clip the objects against the boundary planes of the view volume.

3. To clip a line segment against the view volume, we would need to test the relative position of the line using the view volume's boundary plane equations.

4. By substituting the line endpoint coordinates into the plane equation of each boundary which determines whether the endpoint is inside or outside that boundary.

   a. An endpoint $(x, y, z)$ of a line segment is outside a boundary plane if $Ax + By + Cz + D > 0$

   b. Similarly, the point is inside the boundary if $Ax + By + Cz + D < 0$.

   c. Lines with both endpoints outside a boundary plane are discarded, and those with both endpoints inside all boundary planes are saved.

5. The intersection of a line with a boundary is found using the line equations along with the plane equation.

6. Intersection coordinates $(x_1, y_1, z_1)$ are values that are on the line and that satisfy the plane equation $Ax_1 + By_1 + Cz_1 + D = 0$, where $A$, $B$, $C$, and $D$ are the plane parameters for that boundary.

**Que 3.16.** **Write 3D line clipping algorithm for Cohen-Sutherland region code method.**

**OR**

**Establish and write Cohen and Sutherland 3D line clipping algorithm.** | **AKTU 2012-13, Marks 10**

**Answer**

1. The Cohen Sutherland algorithm extends easily to 3D. It extends the 3D clipping window boundaries to define 27 regions. Assign a 6 bit code to each region, that is, for each point $(x, y, z)$.

2. Suppose the bands for cuboid view volume are $(X_{v\min}, X_{v\max})$, $(Y_{v\min}, Y_{v\max})$ and $(Z_{v\min}, Z_{v\max})$ where the subscript vmin and vmax stands for view volume minlimit and view volume maxlimit (front and back end planes).

3. We assign bit codes to any point $(x, y, z)$ as :

   Bit1 = 1 if $x < X_{\min}$

   Bit 2 = 1 if $x > X_{\max}$

   Bit 3 = 1 if $y < Y_{\min}$

   Bit 4 = 1 if $y > Y_{\max}$

   Bit 5 = 1 if $z < Z_{\min}$

   Bit 6 = 1 if $z > Z_{\max}$.

   Otherwise bits are set to 0 (zero).

4. A point is inside if the region code is 000000. Therefore a line is visible if both ends are 000000.

5. A line is invisible if it lies entirely to one side of the clipping region, that is, if logical AND operation on the bit codes of the end points is non-zero.

**Que 3.17.** **Establish and write Cyrus-Beck 3D line clipping algorithm.** | **AKTU 2013-14, Marks 06**
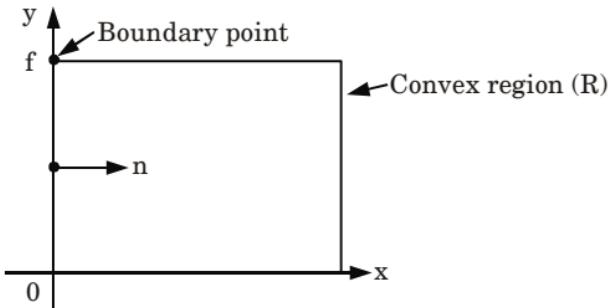
**Answer**

1. Cyrus-Beck algorithm is applicable to an arbitrary convex region.

2. This algorithm uses a parametric equation of line segment to find the intersection points of a line with the clipping edges.

3.  The parametric equation of a line segment from $P_1$ to $P_2$ is

    $$P(t) = P_1 + (P_2 - P_1)\, t, \ 0 \leq t \leq 1$$

    where $t$ is a parameter, $t = 0$ at $P_1$ and $t = 1$ at $P_2$.

4.  Consider a convex clipping region $R$, $f$ is a boundary point of the convex region $R$ and $n$ is an inner normal for one of its boundaries, as shown in Fig. 3.17.1.



**Fig. 3.17.1.** Convex region, boundary point and inner normal.

5.  We evaluate the dot product $n \cdot [P(t) - f]$, and check the following condition as :

    a.  If dot product is negative, *i.e.*,

        $$n \cdot [P(t) - f] < 0$$

        Then the vector $P(t) - f$ is pointed away from the interior of $R$.

    b.  If dot product is zero, *i.e.*,
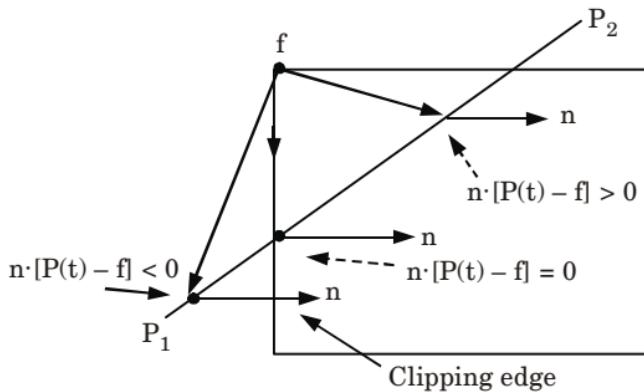
        $$n \cdot [P(t) - f] = 0$$

        Then $P(t) - f$ is pointed parallel to the plane containing $f$ and perpendicular to the normal.

    c.  If dot product is positive, *i.e.*,

        $$n \cdot [P(t) - f] > 0$$

        Then the vector $P(t) - f$ is pointed towards the interior of $R$, as shown in Fig. 3.17.2.



**Fig. 3.17.2.** Dot product for three inside, outside and on the boundary of the clipping region.

**Cyrus-Beck line clipping algorithm :**

1.  Read two endpoints of the line, $P_1$ and $P_2$.

2.  Read vertex coordinates of the clipping window.

3.  Calculate $D = P_2 - P_1$.

4.  Assign boundary point ($f$) with particular edge.

5.  Find inner normal vector for corresponding edge.

6.  Calculate $D \cdot n$ and $W = P_1 - f$.

7.  If $D \cdot n > 0$.

$$t_L = -\frac{W \cdot n}{D.n}$$

else

$$t_U = -\frac{W \cdot n}{D.n}$$

end if

8.  Repeat steps 4 to 7 for each edge of the clipping window.

9.  Find maximum lower limit and minimum upper limit.

10. If maximum lower limit and minimum upper limit do not satisfy condition $0 \le t \le 1$ then ignore the line.

11. Calculate the intersection points by substituting values of maximum lower limit and minimum upper limit in the parametric equation of the line $P_1 P_2$.

12. Draw the line segment $P(t_L)$ to $P(t_U)$.

13. Stop.

**Que 3.18.** Consider the line from $P_1 (-2, 1)$ to $P_2 (8, 4)$ clipped to the rectangular region $R$ as shown in the Fig. 3.18.1. The line $P_1 P_2$ intersects the window. Calculate the intersection points.
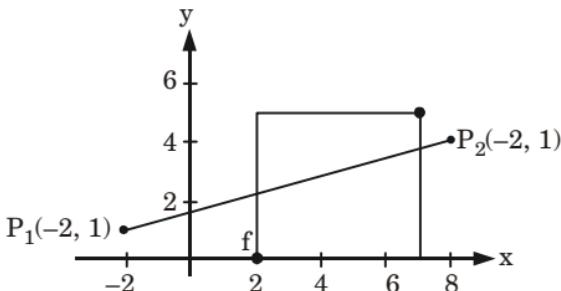


**Fig. 3.18.1.**

**Answer**

1.  The parametric representation of the line $P_1P_2$ is

$$P_{(t)} = P_1 + (P_2 - P_2)t = [-2\ 1] + [10\ 3]\ t$$
$$= (10t - 2)\ i + (3t + 1)\ j \qquad 0 \le t \le 1$$

where $i$ and $j$ are the unit vectors in the $x$ and $y$ directions, respectively.

2.  The four inner normal's are given as

Left :              $n_L = i$

Right :             $n_R = -i$

Bottom :            $n_B = j$

Top :               $n_T = -j$

3.  Choosing $f(2, 0)$ for the left edge gives

$$P(t) - f = (10t - 4)\ i + (3t + 1)\ j \quad \text{and}$$
$$n_L \cdot [P(t) - f] = 10t - 4 = 0$$
$$\therefore \qquad t = 2/5$$

4.  Substituting value of $t$ in parametric equation we get,

$$P(2/5) = [-2\ \ 1] + [10\ \ 3]\ (2/5)$$
$$= [-2\ \ 1] + [4\ \ 6/5]$$
$$= [2\ \ 2.2]$$

5.  Choosing $f(7, 5)$ for the right edge gives

$$P(t) - f = (10t - 9)\ i + (3t - 4)\ j \text{ and}$$
$$n_R \cdot [P(t) - f] = -(10t - 9) = 0$$
$$t = 9/10$$

6.  Substituting value of $t$ in parametric equation we get,

$$P(9/10) = [-\ 2\ \ 1] + [10\ \ 3]\ (9/10)$$
$$= [-2\ \ 1] + [9\ \ 27/10]$$
$$= [7\ \ 37/70] = [7\ \ 3.7]$$

7.  Using $f(2, 0)$ for the bottom edge gives

$$n_B \cdot [P(t) - f] = 3t + 1 = 0$$
$$\therefore \qquad t = -1/3$$

8.  This value of $t$ is outside the range of $0 \leq t \leq 1$ and hence it is rejected. Using $f(7, 5)$ for the top edge gives

$$n_T \cdot [P(t) - f] = -(3t - 4) = 0$$

$\therefore$                          $t = 4/3$

9.  This value of $t$ is outside the range of $0 \leq t \leq 1$ and hence it is rejected.

    Thus, we get two intersection points $(2, 2.2)$ and $(7, 3.7)$ with left edge and right edge, respectively.

☺☺☺

# 4
**UNIT**

# Curves and Surfaces

# CONTENTS

---

PART-1

*Curves and Surfaces : Quadric Surfaces, Spheres, Ellipsoid, Blobby Objects.*

---

CONCEPT OUTLINE

- Curves are the graphical representation of any quantity between $x$ and $y$-axis.
- Ellipsoid surface is the extension of a spherical surface.
- Quadric surface include spheres, ellipsoids, tori, paraboloids, etc.
- Blobby object are non-rigid object that do not have fixed shape and change their surface characteristics in certain motions.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 4.1.** Explain various types of quadratic surfaces with their equation and blobby objects. | AKTU 2014-15, Marks 06
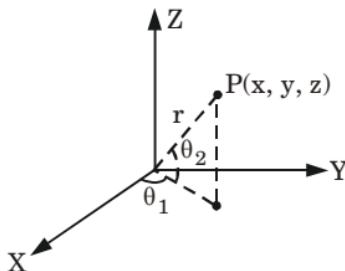
**Answer**

**Various types of quadratic surfaces :**

**1. Sphere :**

a. A sphere with a radius $r$ is defined as the set of points $(x, y, z)$ that satisfies the following equations :

$$x^2 + y^2 + z^2 = r^2$$



**Fig. 4.1.1.** Parametric coordinate position $(r, \theta_1, \theta_2)$ on the surface of a sphere.

b. The parametric equations for spherical surfaces are :
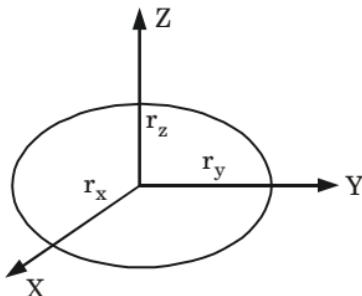
$$x = r \cos \theta_1 \cos \theta_2$$

$$y = r \sin \theta_1 \cos \theta_2$$
$$z = r \sin \theta_2$$

where, $-\dfrac{\pi}{2} \le \theta_2 \le \dfrac{\pi}{2}$ and $-\pi \le \theta_1 \le \pi$

## 2. Ellipsoid :

a. An ellipsoid surface is the extension of a spherical surface. There are three mutually perpendicular radii that have different values as shown in Fig. 4.1.2.



**Fig. 4.1.2.** An ellipsoid with three mutually perpendicular radii $r_x$, $r_y$ and $r_z$, centered on the coordinate origin.

b. In the cartesian coordinate system, the ellipsoid centered at the origin can be defined as :

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

c. The parametric equations are :

$$x = r_x \cos \theta_1 \cos \theta_2$$
$$y = r_y \sin \theta_1 \cos \theta_2$$
$$z = r_z \sin \theta_2$$
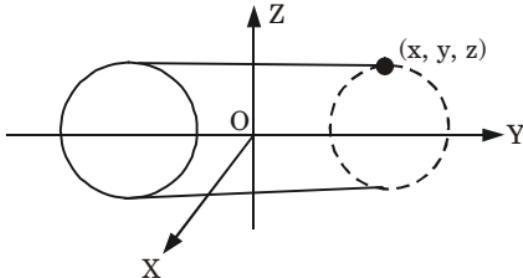
where, $-\pi/2 \le \theta_1 \le \pi/2$ and $-\pi \le \theta_2 \le \pi$

## 3. Torus :

a. A doughnut-shaped object as shown in Fig. 4.1.3 is called a torus.



**Fig. 4.1.3.** Torus

This can be generated by rotating a conic object like a circle about a specified axis.

b. In cartesian coordinates, equations for points over the surface of a torus is given as

$$\left[ r - \sqrt{\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2} \right]^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

c.   The parametric representations for a torus can be given as

$x = r_x(r + \cos\phi) \cos\theta,$ $\qquad\qquad -\pi \leq \phi \leq \pi$

$y = r_y(r + \cos\phi) \sin\theta,$ $\qquad\qquad -\pi \leq \theta \leq \pi$

$z = r_z \sin\phi$

**Blobby objects :**

1.   Blobby objects do not maintain a fixed shape and change their surface characteristics in certain motions or when they come in contact with other objects.

2.   Molecular structures, melting objects, water droplets, muscle shapes in the human body, etc., are examples of blobby objects.

3.   As objects show a certain degree of fluidity and blobbiness, these objects are simply known as blobby objects.



**Fig. 4.1.4.** Molecular bonding as two molecules move away from each other, the surface shape stretch, snap and finally contract into spheres.



**Fig. 4.1.5.** Blobby muscle shape in the human arm.

PART-2

*Introductory Concepts of Spline.*

CONCEPT OUTLINE

•   B-spline is basis spline function which contains a set of control points.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

Que 4.2.   **Explain the B-spline curves. What are the properties of B-spline curve ?**
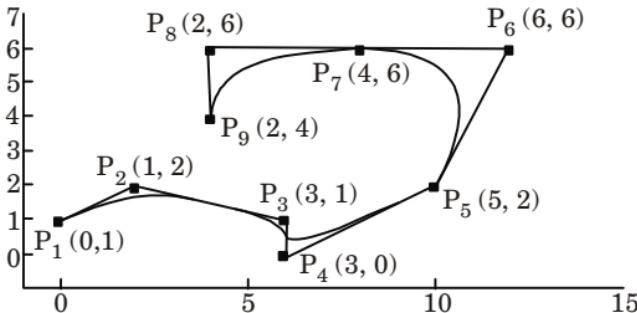
**OR**

**Write the properties of B-spline curves. Also write advantages of B-spline curves over Bezier curves.**

**Answer**

**B-spline curves :**

1. B-spline is basis spline function which contains a set of control points.
2. B-spline function is a combination of flexible band that passes through the number of control points and creates a smooth curve.
3. Each vertex affects the shape of the curve only over a range of parameter values where its associated basis function is non-zero.



**Fig. 4.2.1.**
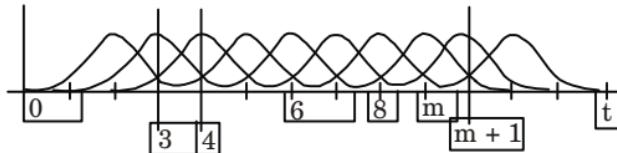
4. The B-spline is non-global because each vertex $B_i$ is associated with a unique basis function.
5. If $P(u)$ be the position vectors along the curve as a function of the parameter $u$, a B-spline curve is given by

$$P(u) = \sum_{k=0}^{n} P_k \, B_{k,d}(u) \, , \, u_{\min} \le u \le u_{\max}$$
$$2 \le d \le n + 1$$

   where the $p_k$ is an input set of $n + 1$ control points.

**Example :** The unweighted spline set of 10 control points, 10 splines, 14 knots, and but only 7 intervals are shown is Fig. 4.2.2.



**Fig. 4.2.2.**

**Properties of B-splines curves :**

1. The polynomial curve has degree $d–1$ and $C^{d–2}$ continuity over the range of $u$.
2. For $n + 1$ control points, the curve is described with $n + 1$ blending functions.
3. Each blending function $B_{k,d}$ is defined over $d$ subintervals of the total range of $u$, starting at knot value $u_k$.

4.  The range of parameter $u$ is divided into $n + d$ subinterval by the $n + d + 1$ values specified in the knot vector.

5.  Each section of spline curve is influenced by $d$ control points.

6.  Any one control point can affect the shape of atmost $d$ curve sections.

7.  For any value of $u$ in the interval from knot value $u_{d-1}$ to $u_{n+1}$ the sum over all basic function is 1 :

$$\sum_{k=0}^{n} B_{k,d}(u) = 1$$

**Advantages of B-spline curves over Bezier curves :**

1.  The degree of B-spline polynomial can be set independently of the number of control points.

2.  B-spline allows local control over the shape of a spline curve or surface.

3.  The trade-off is that B-splines are more complex than Bezier curves.

---

**Que 4.3.** **What are the parametric continuity conditions in a spline representation ?**

**Answer**

We imposed various continuity conditions at the connection points to ensure smooth transition from one section to the next section of the piecewise parametric curve. If the spline curve is described in terms of parametric coordinate form like

$$x = x(v), y = y(v), z = z(v)$$

Spline curve can be represented as :
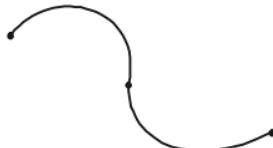
1.  **Zero-order parametric continuity :** When two pieces of the curve meet at a point, this is known as zero-order parametric continuity as shown in Fig. 4.3.1, and is represented by $C^0$.



**Fig. 4.3.1.** Zero-order.

The values of $x, y, z$ can be evaluated at $v$, for the first piece of curve and will be equal to the values of $x, y, z$ evaluated at $v$, for the second curve.

2.  **First-order parametric continuity :** The first-order parametric continuity shows that the first derivatives of the coordinate function $x = x(v), y = y(v), z = z(v)$ for two successive pieces are equal at their joining point as shown in Fig. 4.3.2. This is represented by $C^1$.



**Fig. 4.3.2.** First-order.

3. **Second-order parametric continuity :** The second-order parametric continuity shows that both the first and second-order derivatives of the two curve pieces are the same at intersection as shown in Fig. 4.3.3. This is represented by $C^2$.



**Fig. 4.3.3.** Second-order.

**Que 4.4.** **Explain the characteristics of B-spline curve. How is it useful for interpolation ?**

**Answer**

**Characteristics of B-spline curve :**

1. The sum of the B-spline basis functions for any parameter value is 1.
2. Each basis function is positive or zero for all parameter values.
3. Each basis function has precisely one maximum value, except for $k = 1$.
4. The maximum order of the curve is equal to the number of vertices of defining polygon.
5. The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
6. The curve exhibits the variation diminishing property.
7. The curve generally follows the shape of defining polygon.

**B-spline curve is useful for interpolation because :**

1. The degree of B-spline polynomial is independent of the number of control points of defining polygon.
2. B-spline allows local control over the curve surface because each control point affects the shape of a curve only over a range of parameters values where its associated basis function is non-zero.

**Que 4.5.** **Describe parametric representation of surfaces.**

**Answer**

Parametric representation of surfaces means a continuous, vector-valued function $P(s, t)$ of two variables, or parameters $s$ and $t$ where the variables are allowed to range over connected region of the $uv$ plane.

**Parametric description :**

1. The three equations for determining the coordinates of any point on the surface $S$ are described in terms of parameters, say, $s$ and $f$, and in parameter ranges $[a, b]$ and $[c, d]$, which may be infinite :

$$x = f(s, t), \quad a \le s \le b$$
$$S : y = g(s, t), \quad c \le t \le d$$
$$z = h(s, t)$$

2. The coordinates of any point $P$ on the surface have the form $[f(s, f), g(s, t), h(s, t)]$.

3. Spherical surface in parametric form can be defined in terms of the angular parameters $\phi$ and $\theta$ :

$$x = r \cos \phi \cos \theta, \text{ for } -\pi/2 \le \phi \le \pi/2$$
$$y = r \cos \phi \sin \theta, \text{ for } -\pi \le \theta \le \pi$$
$$z = r \sin \phi$$

4. Parametric representation for a torus is similar to those for an ellipse, except that the angle $\phi$ extends over 360°. Using latitude and longitude angles $\phi$ and $\theta$, we can describe the torus surface as the set of points that satisfy

$$X = r_x \cos \phi \cos \theta, \text{ for } -\pi/2 \le \phi \le \pi/2$$
$$Y = r_y \cos \phi \sin \theta, \text{ for } -\pi \le \theta \le \pi$$
$$Z = r_z \sin \phi$$

---

**Que 4.6.** | **Write note on interpolation.**

**Answer**

1. Interpolation is a method of constructing new data points within the range of a discrete set of known data points.

2. The number of data points obtained by sampling or experimentation represents the values of a function for a limited number of values of the independent variable.

3. It is often required to interpolate, *i.e.*, estimate the value of that function for an intermediate value of the independent variable.

4. A few data points from the original function can be interpolated to produce a simpler function which is still fairly close to the original. The resulting gain in simplicity may outweigh the loss from interpolation error.

5. The main task in this process is to find the suitable mathematical expression for the known curve.

6. Interpolation technique is used when we have to draw the curve by determining intermediate points between the known sample points.

---

**Que 4.7.** | **Explain the types of interpolation method.**

**Answer**

Type of interpolation methods are :

## 1.  IDW :

a.  The IDW (Inverse Distance Weighted) tool uses a method of interpolation that estimates cell values by averaging the values of sample data points in the neighbourhood of each processing cell.

b.  The closer a point is to the center of the cell being estimated the more influence or weight it has in the averaging process.

c.  A specified number of points or all points within a specified radius can be used to determine the output value of each location.

## 2.  Kriging :

a.  Kriging is a geostatistical procedure that considers both the distance and the degree of variation between known data points when estimating values in unknown areas.

b.  This procedure generates an estimated surface from a scattered set of points with $z$-values.

c.  It should be done before we select the best estimation method for generating the output surface.

## 3.  Natural neighbour :

a.  Natural neighbour interpolation finds the closest subset of input samples to a query point and applies weights to them based on proportionate areas to interpolate a value.

b.  It is also known as Sibson or "area-stealing" interpolation.

## 4.  Spline : The spline tool uses an interpolation method that estimates values using a mathematical function that minimizes overall surface curvature, resulting in a smooth surface that passes exactly through the input points.

## 5.  Spline with barriers : The spline with barriers tool uses a method similar to the technique used in the Spline tool, with the major difference being that this tool honors discontinuities encoded in both the input barriers and the input point data.

## 6.  Topo to raster : It use an interpolation technique specifically designed to create a surface that more closely represents a natural drainage surface and better preserves stream networks from input contour data.

## 7.  Trend :

a.  Trend is a global polynomial interpolation that fits a smooth surface defined by a mathematical function (a polynomial) to the input sample points.

b.  The trend surface changes gradually and captures coarse-scale patterns in the data.

---

## PART-3

*B-Spline and Bezier Curve and Surfaces.*

<hr>

## CONCEPT OUTLINE

- A Bezier curve section can be filled to any number of control points.

<hr>
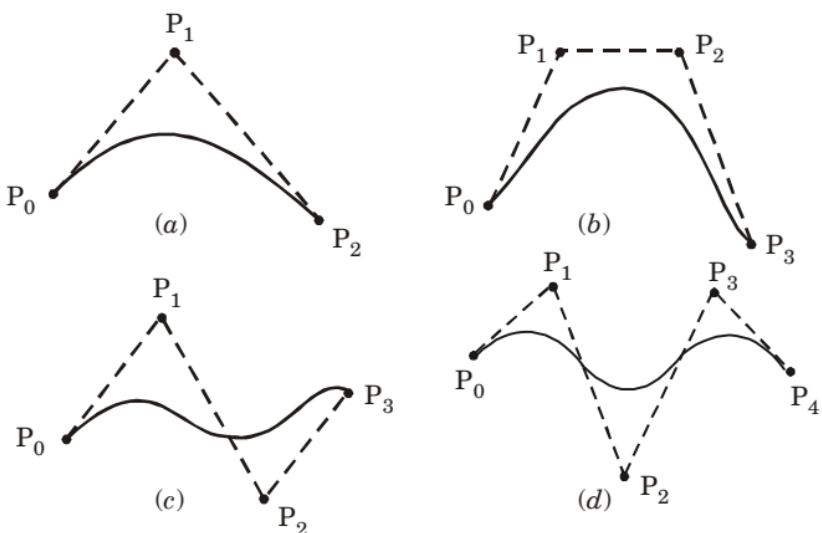
**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

<hr>

**Que 4.8.** | **Explain Bezier curves and properties of Bezier curves.**

**Answer**

**Bezier curves :**

1. A Bezier curve section can be filled to any number of control points.
2. The number of control points to be approximated and their relative position determine the degree of Bezier polynomial.
3. Bezier curve can be specified with boundary conditions with a characterizing matrix or with blending functions.
4. For general Bezier curves, the blending function specification is most convenient.
5. Suppose we are given $n + 1$ control points position : $P_k = (x_k, y_k, z_k)$ with $k$ varying from 0 to $n$.



**Fig. 4.8.1.** Example of Bezier curves generated from three, four, five control points.

6.   These coordinate point can be blended to produce the following position vector $P(u)$, which describes the path of an approximation Bezier polynomial function between $P_0$ and $P_n$.

$$P(u) = \sum_{k=0}^{n} P_k BEZ_{k,n}(u) \ \ 0 \le u \le 1 \qquad \text{...(4.8.1)}$$

Equation (4.8.1) represents a set of three parametric equations for individual coordinates :

$$x(u) = \sum_{k=0}^{n} x_k BEZ_{k,n}(u)$$

$$y(u) = \sum_{k=0}^{n} y_k BEZ_{k,n}(u)$$

$$z(u) = \sum_{k=0}^{n} z_k BEZ_{k,n}(u)$$

where,      $BEZ_{k,n}(u) = c(n, k) \cdot u^k \cdot (1-u)^{n-k}$

**Properties of Bezier curves :**

1.   Bezier curve always passes through the first and last control points *i.e.*, curve has same end points as the guiding polygon.

2.   The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. Therefore, for four control points, the degree of the polynomial is three, *i.e.*, cubic polynomial.

3.   The direction of the tangent vector at the end points is the same as that of the vector determined by first and last segments.

4.   The curve lies entirely within the convex hull formed by four control points.

5.   The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.

6.   The curve is invariant under an affine transformation.

**Que 4.9.**   **Construct the Beizer curve of order 3 with 4 polygon vertices $A(1, 1)$, $B(2, 3)$, $C(4, 3)$, $D(6, 4)$.**

**Answer**

The equation for the Bezier curve is given as :

$\quad p(u) = (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2(1-u) P_3 + u^3 P_4$ for $0 \le u \le 1$

where $p(u)$ is the point on the curve $P_1, P_2, P_3, P_4$.

Let us take        $u = 0, \dfrac{1}{4}, \dfrac{1}{2}, \dfrac{3}{4}$

$\therefore$              $P(0) = (1, 1)$

$$\therefore \quad P\left(\frac{1}{4}\right) = \left(1 - \frac{1}{4}\right)^3 P_1 + 3\frac{1}{4}\left(1 - \frac{1}{4}\right)^2 P_2$$
$$+ 3\left(\frac{1}{4}\right)^2 \left(1 - \frac{1}{4}\right) P_3 + \left(\frac{1}{4}\right)^3 P_4$$

$$= \frac{27}{64}(1, 1) + \frac{27}{64}(2, 3) + \frac{9}{64}(4, 3) + \frac{1}{64}(6, 4)$$

$$= \left[\frac{27}{64} \times 1 + \frac{27}{64} \times 2 + \frac{9}{64} \times 4 + \frac{1}{64} \times 6, \frac{27}{64} \times 1\right.$$
$$\left. + \frac{27}{64} \times 3 + \frac{9}{64} \times 3 + \frac{1}{64} \times 4\right]$$

$$= \left[\frac{123}{64}, \frac{139}{64}\right] = (1.9218, 2.1718)$$

$$\therefore \quad P\left(\frac{1}{2}\right) = \left(1 - \frac{1}{2}\right)^3 P_1 + 3\frac{1}{2}\left(1 - \frac{1}{2}\right)^2 P_2$$
$$+ 3\left(\frac{1}{2}\right)^2 \left(1 - \frac{1}{2}\right) P_3 + \left(\frac{1}{2}\right)^3 P_4$$

$$= \frac{1}{8}(1, 1) + \frac{3}{8}(2, 3) + \frac{3}{8}(4, 3) + \frac{1}{8}(6, 4)$$

$$= \left[\frac{1}{8} \times 1 + \frac{3}{8} \times 2 + \frac{3}{8} \times 4 + \frac{1}{8} \times 6, \frac{1}{8} \times 1\right.$$
$$\left. + \frac{3}{8} \times 3 + \frac{3}{8} \times 3 + \frac{1}{8} \times 4\right]$$

$$= \left[\frac{25}{8}, \frac{23}{8}\right] = (3.125, 2.875)$$

$$\therefore \quad P\left(\frac{3}{4}\right) = \left(1 - \frac{3}{4}\right)^3 P_1 + 3\frac{3}{4}\left(1 - \frac{3}{4}\right)^2 P_2$$
$$+ 3\left(\frac{3}{4}\right)^2 \left(1 - \frac{3}{4}\right) P_3 + \left(\frac{3}{4}\right)^3 P_4$$

$$= \frac{1}{64} P_1 + \frac{9}{64} P_2 + \frac{27}{64} P_3 + \frac{27}{64} P_4$$

$$= \frac{1}{64}(1, 1) + \frac{9}{64}(2, 3) + \frac{27}{64}(4, 3) + \frac{27}{64}(6, 4)$$

$$= \left[\frac{1}{64} \times 1 + \frac{9}{64} \times 2 + \frac{27}{64} \times 4 + \frac{27}{64} \times 6, \frac{1}{64} \times 1\right.$$
$$\left. + \frac{9}{64} \times 3 + \frac{27}{64} \times 3 + \frac{27}{64} \times 4\right]$$

$$= \left[\frac{289}{64}, \frac{217}{64}\right] = (4.5156, 3.3906)$$

$$P(1) = P_3 = (6, 4)$$

Fig. 4.9.1 shows the calculated points of the Bezier curve and curve passing through it.



**Fig. 4.9.1.** Plotted Bezier curve.

**Que 4.10.** Construct enough points on the Bezier curve whose control points are $p_0(4, 2)$, $p_1(8, 8)$ and $p_2(16, 4)$ to draw an accurate sketch.

1. What is the degree of the curve ?
2. What are coordinates at $u = 0.5$.

**Answer**

1. Here $n = 2$ *i.e.*, order = 2

   $$\text{Degree} = \text{order} - 1 = 2 - 1 = 1$$

2. Blending function of the curve :

   $$B_{0,2}(u) = {}^2C_0 \, u^0 \, (1-u)^{2-0} = (1-u)^2$$
   $$B_{1,2}(u) = {}^2C_1 \, u^1 \, (1-u)^{2-1} = 2u(1-u)$$
   $$B_{2,2}(u) = {}^2C_2 \, u^2 \, (1-u)^{2-2} = u^2$$

   then Bezier curve for given control points :

   $$C(u) = p_0 \, B_{0,2}(u) + p_1 \, B_{1,2}(u) + p_2 \, B_{2,2}(u)$$

   Bezier curve in $x$ and $y$ coordinate is given as :

   $$x(u) = x_0 \, B_{0,2}(u) + x_1 \, B_{1,2}(u) + x_2 \, B_{2,2}(u)$$
   $$y(u) = y_0 \, B_{0,2}(u) + y_1 \, B_{1,2}(u) + y_2 \, B_{2,2}(u)$$

   Put     $x_0 = 4, y_0 = 2$
   $$x_1 = 8, y_1 = 8$$
   $$x_2 = 16, y_2 = 4$$

$$x(u) = 4(1 - u)^2 + 8 \times 2u(1 - u) + 16u^2$$
$$= 4u^2 + 8u + 4$$

at $\quad\quad\quad\quad u = 0.5$

$$x(u) = 9$$

$$y(u) = 2(1 - u)^2 + 8 \times 2u(1 - u) + 4u^2$$
$$= -10u^2 + 12u + 2$$

at $\quad\quad\quad\quad u = 0.5$

$$y(u) = 5.5$$

Therefore, the coordinates at $u = 0.5$ are (9, 5.5).

**Que 4.11.** **Write at least four properties of Bezier curves. Calculate and also, roughly trace the Bezier curve for three control points (1, 1), (2, 2) and (3, 1).**   **AKTU 2012-13, 2013-14; Marks 06**

**Answer**

**Properties of Bezier curves :** Refer Q. 4.8, Page 4–10C, Unit-4.

**Numerical :**

Control functions for point (1, 1), (2, 2), (3, 1) are :

$$B_{0,2}(u) = {}^2C_0\, u^0\, (1 - u)^{2-0}$$
$$= (1 - u)^2$$
$$B_{1,2}(u) = {}^2C_1\, u^1(1 - u)^{2-1}$$
$$= 2u(1 - u)$$
$$B_{2,2}(u) = {}^2C_2\, u^2\, (1 - u)^{2-2}$$
$$= u^2$$

Bezier curve, $\quad\quad C(u) = p_0 B_{0,2}(u) + p_1 B_{1,2}(u) + p_2 B_{2,2}(u)$

Dissolving curve in the $x$, $y$ coordinate :

$$x(u) = x_0\, B_{0,2}(u) + x_1 B_{1,2}(u) + x_2\, B_{2,2}(u)$$
$$y(u) = y_0 B_{0,2}(u) + y_1 B_{1,2}(u) + y_2 B_{2,2}(u)$$

where, $\quad\quad\quad x_0 = 1$ $\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad y_1 = 1$

$\quad\quad\quad\quad\quad\quad x_1 = 2$ $\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad y_2 = 2$

$\quad\quad\quad\quad\quad\quad x_2 = 3$ $\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad y_3 = 1$

$$x(u) = 1(1 - u)^2 + 2[2u(1 - u)] + 3u^2$$
$$= 1 + u^2 - 2u + 4u - 4u^2 + 3u^2 = 1 + 2u$$
$$y(u) = 1\,(1 - u)^2 + 2 \times [2u(1 - u)] + 1u^2$$
$$= 1 + u^2 - 2u + 4u - 4u^2 + u^2 = 1 + 2u - 2u^2$$

**Fig. 4.11.1.**

**Que 4.12.** Compare and contrast among spline, B-spline and Bezier algorithms for curve generation and write the algorithm for Bezier curve generation.

AKTU 2015-16, Marks 15

**Answer**

**Algorithm for Bezier curve :**

1. Get four control points say A $(x_A, y_A)$, B$(x_B, y_B)$, C$(x_C, y_C)$, D$(x_D, y_D)$.

2. Divide the curve represented by points $A$, $B$, $C$ and $D$ in two sections

$$x_{AB} = (x_A + x_B) / 2$$
$$y_{AB} = (y_A + y_B) / 2$$
$$x_{BC} = (x_B + x_C) / 2$$
$$y_{BC} = (y_B + y_C) / 2$$
$$x_{CD} = (x_C + x_D) / 2$$
$$y_{CD} = (y_C + y_D) / 2$$
$$x_{ABC} = (x_{AB} + x_{BC}) / 2$$
$$y_{ABC} = (y_{AB} + y_{BC}) / 2$$
$$x_{BCD} = (x_{BC} + x_{CD}) / 2$$
$$y_{BCD} = (y_{BC} + y_{CD}) / 2$$
$$x_{ABCD} = (x_{ABC} + x_{BCD}) / 2$$
$$y_{ABCD} = (y_{ABC} + y_{BCD}) / 2$$

3. Repeat the step 2 for section $A$, $AB$, $ABC$ and $ABCD$ and section $ABCD$, $BCD$, $CD$ and $D$.

4. Repeat step 3 until we have sections so short that they can be replaced by straight lines.

5. Replace small sections by straight lines.

6. Stop.

**Comparison :**

| S. No. | Spline | B-spline | Bezier |
|--------|--------|----------|--------|
| 1. | A spline curve can be specified by giving a set of coordinate positions, called control points which indicates the general shape of the curve. | The B-spline curves are specified by a Bernstein basis function that has a limited flexibility. | The Bezier curves can be specified with boundary conditions, with a characterizing matrix or with blending functions. |
| 2. | The curve follows the general shape of the curve. | The curves resulted by the use of open uniform basis function are nearly like Bezier curves. | The curve generally follows the shape of defining polygon. |
| 3. | Typical CAD application for splines include the design of automobile bodies, aircraft and space craft surfaces and ship hulls. | These curves can be used to construct blending curves. | Bezier curves are found in painting & drawing packages as well as CAD system. |
| 4. | It possess a high degree of smoothness at the places where the polynomial pieces connect. | The B-spline allows the order of the basis function and hence the degree of the resulting curve is independent on the number of vertices. | The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. |

---

**Que 4.13.** Let $P_0(0, 0)$, $P_1(1, 2)$, $P_2(2, 1)$, $P_3(3, 1)$, $P_4(4, 10)$ and $P_5(5, 5)$ be given data control points. If interpolation based on Bezier curve is used to find a curve interpolating these data points. Find parametric midpoint of the gradient and also calculate coordinate of parametric quartiles of the curve.         **AKTU 2014-15, Marks 06**
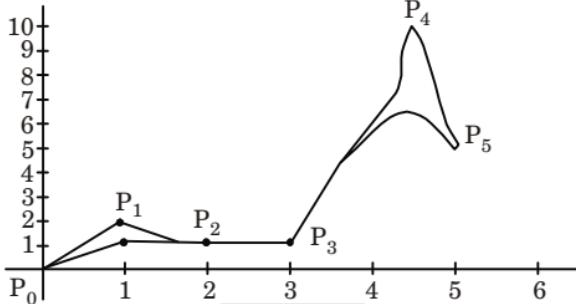
**Answer**

Given points : $P_0(0, 0)$, $P_1(1, 2)$, $P_2(2, 1)$, $P_3(3, 1)$, $P_4(4, 10)$ and $P_5(5, 5)$
$$u = 0.5$$

$$n + 1 = 6$$
$$n = 5$$

$$P(u) = \sum_{k=0}^{5} P_k \, BEZ_{k,n}(u)$$

$$BEZ_{k,5}(u) = C(5, k) \, u^k \, (1 - u)^{5-k}$$



**Fig. 4.13.1.**

$$C_{5,k} = \frac{\lfloor 5}{\lfloor k \, \lfloor 5-k}$$

$$B_{0,5}(u) = C_{5,0} \, u^0 (1-u)^{5-0} = (1-u)^5$$
$$B_{1,5}(u) = C_{5,1} \, u^1 (1-u)^4 = 5(u) \, (1-u)^4$$
$$B_{2,5}(u) = C_{5,2} \, u^2 (1-u)^3 = 10u^2 \, (1-u)^3$$
$$B_{3,5}(u) = C_{5,3} \, u^3 (1-u)^2 = 10u^3 \, (1-u)^2$$
$$B_{4,5}(u) = C_{5,4} \, u^4 (1-u)^1 = 5u^4 \, (1-u)$$
$$B_{5,5}(u) = C_{5,5} \, u^5 (1-u)^0 = u^5$$

$$C_{5,1} = \frac{\lfloor 5}{\lfloor 1 \, \lfloor 4} = \frac{5 \times \lfloor 4}{\lfloor 1 \, \lfloor 4} = 5$$

$$C_{5,2} = \frac{\lfloor 5}{\lfloor 2 \, \lfloor 3} = \frac{5 \times \lfloor 4}{\lfloor 2 \, \lfloor 3} = 5 \times 2 = 10$$

$$C_{5,3} = \frac{\lfloor 5}{\lfloor 3 \, \lfloor 2} = 10$$

$$C_{5,4} = \frac{\lfloor 5}{\lfloor 4 \, \lfloor 1} = 5$$

at $u = 0.5$

$$X(0.5) = 0 + 1 \times 5(0.5) \, (0.5)^4 + 2 \times 10 \times (0.5)^2 \, (0.5)^3 + 3 \times$$
$$10(0.5)^5 + 4 \times 5(0.5)^5 + 5 \times (0.5)^5$$
$$= (0.5)^5 \, (5 + 20 + 30 + 25) = 2.5$$

$$Y(0.5) = 0 + 2^5 \times (0.5)^5 + 1 \times 10(0.5)^5 + 1 \times 10(0.5)^5$$
$$+ 10 \times 5(0.5)^5 + 5 \times (0.5)^5$$
$$= (0.5)^5 \, (10 + 10 + 10 + 50 + 5)$$
$$= 85 \times (0.5)^5 = 2.6562$$

at          $u = 0.5, (X, Y) = (2.5, 2.6562).$

☺☺☺

# 5 UNIT

# Hidden Lines and Surfaces

# CONTENTS

PART-1

*Hidden Lines and Surfaces : Back-Face Detection Algorithm, Depth Buffer Method, A-Buffer Method, Scanline Method.*

CONCEPT OUTLINE

- Surfaces that are not seen by the naked eyes are known as hidden surfaces.
- Back-face detection algorithm is used to detect hidden surface.
- Depth buffer is also known as $z$-buffer which is used to manage coordinates of image depth in 3D.
- A-buffer is known as anti-aliased.
- Scanline rendering is an algorithm for visible surface determination.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

Que 5.1. | Discuss back-face detection method.

**OR**

Write back-face detection algorithm of hidden lines.

AKTU 2012-13, Marks 06

**Answer**

**Back-face detection method :**

1. Object surfaces that are orientated away from the viewer are called back-faces.

2. Algorithm used to detect the back-faces is known as back-face detection algorithm.

3. If any three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$ on any plane surface are known, the unknown parameters $A$, $B$, $C$ and $D$ of the plane surface equation $Ax + By + Cz + D = 0$ can be found as follows :

   i.   All the three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$ should satisfy the equation $Ax + By + Cz + D = 0$ as it lies on the surface.

   Hence,     $Ax_1 + By_1 + Cz_1 + D = 0$

   $Ax_2 + By_2 + Cz_2 + D = 0$

   $Ax_3 + By_3 + Cz_3 + D = 0$

ii.   Also, any arbitrary point $(x, y, z)$ lying on the surface should satisfy the equation of the desired surface

$$Ax + By + Cz + D = 0$$

iii.  A unique solution of equations for $A, B, C$ and $D$ can only be obtained if

$$\text{or} \quad \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x & y & z & 1 \end{vmatrix} = 0 \qquad \text{(By Cramer's rule)}$$

$$\text{or} \quad \begin{vmatrix} x - x_1 & y - y_1 & z - z_1 & 0 \\ x_1 - x_2 & y_1 - y_2 & z_1 - z_2 & 0 \\ x_2 - x_3 & y_2 - y_3 & z_2 - z_3 & 0 \\ x & y & z & 1 \end{vmatrix} = 0$$

$$\text{or} \quad \begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ x_2 - x_3 & y_2 - y_3 & z_2 - z_3 \end{vmatrix} = 0$$

$$\text{or} \quad (x - x_1) \begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} - (y - y_1) \begin{vmatrix} x_1 - x_2 & z_1 - z_2 \\ x_2 - x_3 & z_2 - z_3 \end{vmatrix}$$
$$+ (z - z_1) \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} = 0$$

$$\text{or} \quad \begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} x + \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_1 - z_3 & x_2 - x_3 \end{vmatrix} y$$
$$+ \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0$$

which is in the form of $Ax + By + Cz + D = 0$, giving the equation of the plane passing through the three points

where,
$$A = \begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix}$$

$$B = \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_2 - z_3 & x_2 - x_3 \end{vmatrix}$$

$$C = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix}$$

and
$$D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

iv. Any plane with its inside and outside surfaces can be defined by three points and its normal vector.

v. This normal vector can be calculated with three position vectors of three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$.

vi. Using right hand thumb rule, the three points are considered in counter clockwise manner for calculating the unit normal directing outside the surface. Thus, the surface normal and equation of the plane are calculated.

vii. Now, if any point $(x, y, \text{ or } z)$ lies inside the polygon surface, $Ax + By + Cz + D < 0$, and if it is along the line of sight to the surface, the polygon must be back-face.

viii. If $L$ is the line of sight or viewing vector (light vector in case of shadowing), perpendicular to the viewing plane originating from the eye or camera, $N$ is the unit surface normal vector as shown in Fig. 5.1.1(a) and (b) the polygon is the back-face if $L.N > 0$
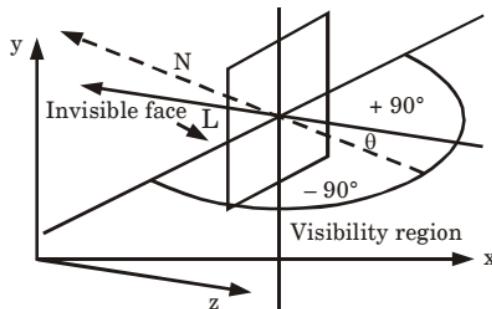


**Fig. 5.1.1.** (a) Geometry of camera vision.

ix. On aligning the light vector to that of the view vector which is normally along negative $z$-axis for right-handed viewing system, that is

$$L = 0i + 0j + L_z k$$
$$L.N = L_z K$$

x. Hence, only $z$-component of normal vector $N$ is required to be considered, and the sign of $K$ is checked for negative value.

xi. Thus, any polygon is a back-face if it has a surface normal with negative $z$-component value.



**Fig. 5.1.1.** (b) Invisible and visible vision.
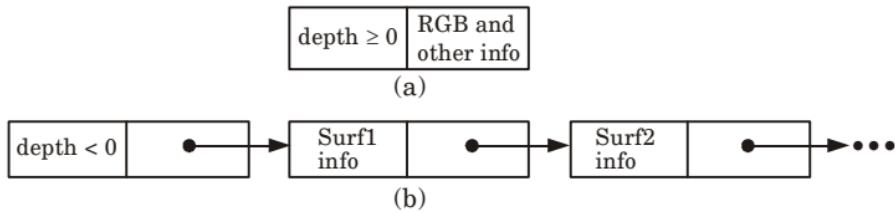
**Que 5.2.** | **Discuss *z*-buffer algorithm.**

**OR**

**Write Depth-buffer method algorithm of hidden lines.**

**AKTU 2013-14, Marks 06**

**Answer**

***z*-buffer/Depth-buffer algorithm :**

1. *z*-buffer is a simple extension of the frame buffer idea. A frame is used to store the intensity of each pixel in image space.

2. The *z*-buffer is a separate depth buffer used to store the *z*-coordinates or depth or every visible pixel in image space.

3. In this algorithm a buffer of the same size as the frame buffer, is set up which holds depth information.

4. Each elements of the depth buffer corresponds to a pixel in the frame buffer and initially holds the maximum depth in the scene.

5. As each polygon is scan-converted, the depth at each pixel is calculated and compared with the corresponding depth in the depth buffer.

6. If the depth is less than that stored in the depth buffer (*i.e.*, nearer the viewer) then that pixel is set in the frame buffer with the polygon colour at that point and the depth buffer is set to the polygon depth.

7. If the polygon depth is greater (*i.e.*, farther away from the viewer) than the depth buffer at that point is not stored in the frame buffer.

8. The *z*- buffer algorithm can be stated as :

   **Step 1 :** Initialize frame buffer to background colour.

   **Step 2 :** Initialize *z*-buffer to minimum *z* value.

   **Step 3 :** Scan-convert each polygon in arbitrary order.

   **Step 4 :** For each $(x, y)$ pixel, calculated depth '*z*' at that pixel ($z(x, y)$).

   **Step 5 :** Compare calculated new depth $z(x, y)$ with value previously stored in *z*-buffer at the location $z(x, y)$.

   **Step 6 :** If $z(x, y) > z(x, y)$, then write the new depth value to *z*-buffer and updated frame buffer.

   **Step 7 :** Otherwise, no action is taken.

**Que 5.3.** | **Discuss A-buffer method.**

**Answer**

1. A-buffer method is a visibility detection method for the rendering system.

2. The A-buffer expands on the depth buffer method to allow transparency. The key data structure in the A-buffer is the accumulation buffer.

3. Each position in the A-buffer has two fields :

   **a. Depth field :** It stores a positive or negative real number.

   **b. Intensity field :** It stores surface intensity information or a pointer value.

4. If depth >= 0, the number stored at that position is the depth of a single surface overlapping the corresponding pixel area.

5. The intensity field then stores the RGB components of the surface colour at that point and the percent of pixel coverage.



(a)

(b)

**Fig. 5.3.1.**

6. If depth < 0, it indicates multiple surface contributions to the pixel intensity.

7. The intensity field then stores a pointer to a linked list of surface data.

8. The surface buffer in the A-buffer includes :

   a. RGB intensity components

   b. Opacity parameter

   c. Depth

   d. Percent of area coverage

   e. Surface identifier

9. The depth and opacity values are used to determine the final colour of a pixel.

**Que 5.4.** **List the advantages and disadvantages of back-face detection and A-buffer method. Write the algorithm for back-face detection.** **AKTU 2015-16, Marks 15**

**Answer**

**Advantages of back-face detection method :**

1. It is fast.

2. It is a simple object space method.

**Disadvantages of back-face detection method :**

1. It can only be used on solid objects modeled as a polygon mesh.

2. It works fine for convex polyhedra but not necessarily for concave polyhedra.

**Advantages of A-buffer method :**

1. It is an extension of depth-buffer method.

2. More than one surface intensity can be taken into consideration at each pixel position.

3. It also anti-aliased the object edges.

4. In this, surfaces are sub-divided into a polygon mesh and clipped against the pixel boundaries.

5. The intensity of pixel is determined by considering opacity parameter and percentage of overlaps of the overlapping surface.

**Disadvantages of A-buffer method :**
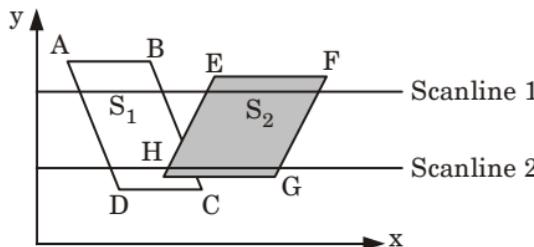
1. This algorithm processes multiple objects at a time. The total number of polygons in a picture can be arbitrarily large.

**Back-face detection algorithm :** Refer Q. 5.1, Page 5–2C, Unit-5.

**Que 5.5.** | **Explain hidden surface removing of scanline method.**

**Answer**

1. A scanline method of hidden surface removal is another approach of image space method.

2. It is an extension of the scanline algorithm, for filling polygon interiors.

3. This algorithm deals with multiple surfaces.

4. As each scanline is processed, all polygon surfaces intersecting that line are examined to determine which are visible.



**Fig. 5.5.1.** Illustration of scanline method of hidden surface removal.

5. Across each scanline, depth calculations are made for each overlapping surface to determine which is nearest to the view plane.

6. When the visible surface has been determined, the intensity value for that position is entered into the frame buffer.

7. This algorithm maintains three table :

   **a. Edge Table (ET) :** It contain the list of edge used in scanline algorithm.

ET entry | X | $y_{max}$ | $\Delta x$ | ID | • ➔

**Fig. 5.5.2.**

b. **Active Edge Table (AET) :** It contains only edges that cross the current scanline, stored in order of increasing $x$.

**AET contents**

| Scanline | Entries |
|----------|---------|
| Scanline 1 | AD  BC  EH  FG |
| Scanline 2 | AD  EH  BC  FG |

c. **Polygon Table (PT) :** Polygon Table (PT) that contains at least the following information for each polygon, in addition to ID :

PT entry | ID | Plane eq. | Shading info. | In-out
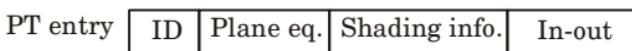
**Fig. 5.5.3.**

i. The coefficient of the plane equation.

ii. Shading or colour information for the polygon.

iii. An in-out Boolean flag, initialized to false and used during scanline processing.

8. In addition, we define a flag for each surface that is set on or off to indicate whether a position along a scanline is inside or outside of the surface. Scanline is processed from left to right.

9. At the leftmost boundary of a surface, the surface flag is turned on, and at the rightmost boundary, it is turned off.

**Que 5.6.** | **Write a short note on visible surface detection or hidden surface elimination.**

**Answer**

1. Visible surface detection or hidden surface elimination is the procedure that removes any surfaces or lines which are not to be displayed in a 3D scene.

2. A process may be developed which draws all parts of all objects in the scene, starting from the object which is farthest from the viewing plane to the nearest one.

3. When we view a picture containing non transparent objects and surfaces, then we cannot see those objects that are viewed from behind the object. We must remove these hidden surfaces to get realistic screen image.

4. The identification and removal of these surfaces is called the hidden-surface problem.

5. Two approaches for visible surface detection are :

    **a.**   **Object space method :** It compares objects and parts of objects to each other within the scene definition to determine which surfaces, as a whole, we should label as visible.

    **b.**   **Image space method :** Visibility of objects is decided point by point at each pixel position on the projection plane.

6.   Scanline coherence method solves the hidden surface problem one scanline at a time from top to bottom.

7.   The simplest scanline algorithm is a one dimensional version of the depth buffer.

8.   In each scanline, depth calculations are made for each overlapping surface to determine which is nearest to the view plane.

---

### PART-2

*Basic Illumination Models-Ambient Light, Diffuse Reflection, Specular Reflection and Phong Model, Combined Approach, Warn Model.*

---

### CONCEPT OUTLINE

- An illumination model is used to calculate the intensity of light that we should see at a given point on the surface of an object.
- Ambient light has no spatial or directional characteristics.
- Diffuse reflection is constant over each surface in a scene.
- Hidden surface methods can be used to locate areas where light sources produce shadows.

---

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

---

**Que 5.7.** | **Explain the basic illumination models.**

**Answer**

1.   An illumination model is needed to calculate the intensity of light that we should see at a given point on the surface of an object.

2.   An illumination model is also known as lighting model or shading model.

**Various illumination models :**

**1.**   **Ambient light :**

    i.   In this model, the intensity of the reflected light for each surface depends on the optical properties of the surface.

ii. Ambient light has no spatial or directional characteristics.

iii. The amount of ambient light incident on each object surface is constant for all surfaces and over all directions.

iv. The optical properties determine how much of the incident energy is to be reflected and how much absorbed.

## 2. Diffuse reflection :

i. In this model, light coming from all directions is reflected from the walls, floor, ceiling, etc.

ii. Diffuse reflections are constant over each surface in a scene, independent of the viewing direction.

iii. The amount of diffused reflected light for each surface in a scene can be set with parameter $K_d$.

iv. For the surfaces which absorbs most of the incident light (black surfaces), the value of the diffuse reflection coefficient ($K_d$) is close to 0.

v. If a surface is exposed to only ambient light, then the intensity of the diffuse reflection at any point on the surface is given by

$$I_{ambdiffuse} = K_d.I_a$$



(a) Radiant energy from surface.

(b) Diffuse reflection from surface.

**Fig. 5.7.1.**

## 3. Phong model :

i. According to the Phong model, the intensity of specular reflection is proportional to $\cos^{n_s} \theta$. The value of $n$, specular reflection parameter, is determined by the type of surface.

ii. For very shiny surfaces, the value of $n_s$ is large (more than 100) and for dull surfaces the value of $n_s$ is close to 1. For a perfect reflector, the value of $n_s$ is infinite.

iii. The intensity of specular reflection depends on the angle of incidence, on the properties of the surface, colour of incident light and polarization.

iv. The specular reflection coefficient $W(\theta)$, for each surface may be used to model the monochromatic specular intensity variation.

v. We can write the Phong specular reflection model as

$$I_{spec} = W(\theta) \, I_l \cos^{n_s} \theta \qquad ...(5.7.1)$$

where $I_l$ is intensity of light source, and $\theta$ is the viewing angle relative to specular reflection direction $R$.

vi. In the case of opaque materials, the specular reflection rejection is almost constant for all incidence angles. Therefore, replacing $W(\theta)$ with a constant specular reflection coefficient $K_s$. Since $V$ and $R$ are unit vectors in the viewing and specular reflection directions, $\cos \theta = V.R$. We can rewrite the equation (5.7.1) as

$$I_{spec} = K_s I_l (V.R)^{n_s}$$



**Fig. 5.7.2.** Projection of L on to the direction of normal vector N.

**Que 5.8.** | **Explain specular reflection and Phong model.**
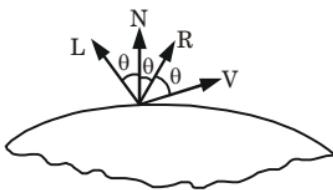
AKTU 2012-13, Marks 10

AKTU 2013-14, Marks 06
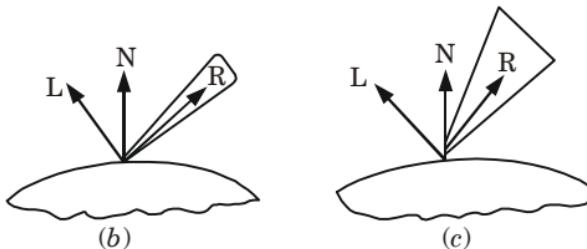
**Answer**

**Specular reflection :**

1.  Specular reflection is the result of total or near total reflection of the light in a concentrated region around the specular reflection angle.

2.  When we see an illuminated shiny surface, like polished metal, shiny apple, or a pearl, then we observe a bright spot or a highlighted spot at a certain viewing direction.

3.  In the Fig. 5.8.1($a$) the specular reflection angle, denoted by $\theta$ is the same as the angle of incident light and $N$ is the unit normal surface vector, $R$ is the unit vector in the direction of ideal specular reflection, $L$ is the unit vector directed towards the point light source, and $V$ is the unit vector pointing towards the viewer from the surface position.

4.  Angle $\theta$ is the viewing angle relative to the specular reflection direction $R$.

5.  In the case of an ideal reflector or perfect mirror, the incident light is reflected only in the specular reflection direction.

6.  In this case, we would only see reflected light when vectors $V$ and $R$ coincide, *i.e.*, $\theta = 0$. The objects other than ideal reflectors have specular reflection around vector $R$.

**Fig. 5.8.1.** (a) Specular reflection angle = Angle of incidence.



(b)                                    (c)

**Fig. 5.8.1.** (b) Shiny (c) Dull surface.

**Phong model :** Refer Q. 5.7, Page 5–9C, Unit-5.

Que 5.9. | **Explain the different illumination methods and different rendering methods in detail. Write its advantages and disadvantages.**

**AKTU 2015-16, Marks 15**

**Answer**

**Various illumination methods :** Refer Q. 5.7, Page 5–9C, Unit-5.

**Advantages of illumination methods :**

1. Fast
2. Acceptable results
3. Hardware support

**Disadvantages of illumination methods :**

1. Point light source
2. No interaction between objects
3. Adhoc, not based on model of light propagation

**Different rendering methods :**

**1. Constant-intensity shading :**

   a. In this method, a single intensity is calculated for each polygon and all points in the surface of polygon are displayed with the same intensity value.

   b. It is fast and simplest rendering method.

   c. Light sources illuminating the object and the viewing position is sufficiently far from the surface of polygon.

2. **Gouraud shading :**

   a. It represents a polygon surface by linearly interpolating intensity across the polygon surface.

   b. Intensity values for each polygon are matched with adjacent polygon along common edges.

   c. It eliminates the intensity discontinuity that can occur in constant-intensity shading.

3. **Phong shading :**

   a. Phong shading is method which interpolates normal vectors and then apply the illumination model to each surface point.

   b. This method interpolates the surface normal vector, instead of the intensity.

   c. It is used in 3D system.

4. **Fast Phong shading :**

   a. Fast Phong shading approximates the intensity calculations using a Taylor-series expansion and triangular surface patches.

   b. Surface rendering with Phong shading can be speeded up by using approximations in the illumination model calculations of normal vectors.

**Advantages of rendering methods :**

i. It removes the intensity discontinuity.

ii. It can be combined with hidden surface algorithm to fill in the visible polygons along each scanline.

iii. It displays more realistic highlights on a surface.

iv. It gives more accurate results.

**Disadvantages of rendering methods :**

i. It has a problem with specular reflections.

ii. It introduces anomalies known as Mach bands.

iii. It requires more calculations and increases the cost of shading steeply.

**Que 5.10.** Discuss combined diffuse and specular reflections with multiple light sources.

**Answer**

1. For a single point light source, we can model the combined diffuse and specular reflections from a point on an illuminated surface as
$$I = I_{diff} + I_{spec}$$
$$= K_a I_a + K_d I_l (N.L) + K_s I_l (N.H)^{n_s}$$

2. If we place more than one point source in a scene, we obtain the light reflection at any surface point by summing the contributions from the individual sources :

$$I = K_a I_a + \sum_{i=1}^{n} I_{li}[K_d(N.L_i) + K_s(N.H_i)^{n_s}]$$

3. To ensure that any pixel intensity does not exceed the maximum allowable value, we can apply some type of normalization procedure.

4. A simple approach is to set a maximum magnitude for each term in the intensity equation.

5. If any calculated term exceeds the maximum, we simply set it to the maximum value.

6. Another way to compensate for intensity overflow is to normalize the individual terms by dividing each by the magnitude of the largest term.

7. A more complicated procedure is first to calculate all pixel intensities for the scene, then the calculated intensities are scaled onto the allowable intensity range.

---

**Que 5.11.** | **Discuss Warn model.**

**OR**

**Explain Phong and Warn model for illumination.**

**AKTU 2014-15, Marks 06**

**Answer**

**Phong model :** Refer Q. 5.7, Page 5–9C, Unit-5.

**Warn model :**

1. The Warn model provides a method for simulating studio lighting effects by controlling light intensity in different directions.

2. Light sources are modeled as points on a reflecting surface, using the Phong model for the surface points. Then, the intensity in different directions is controlled by selecting values for the Phong exponent.

3. In addition, light controls, such as "barn doors" and spotlighting, used by studio photographers can be simulated in the Warn model.

4. Flaps are used to control the amount of light emitted by a source in various directions.

5. Two flaps are provided for each of the $x$, $y$ and $z$-directions.

6. Spotlights are used to control the amount of light emitted within a cone with apex at a point-source position.

---

**PART-3**

*Intensity Attenuation, Colour Consideration,
Transparency and Shadows.*

## CONCEPT OUTLINE

- Intensity attenuation refers to any reduction in the strength of a signal.
- A transparent surface produces both reflected and transmitted light.
- A shadowed object is one which is hidden from the light source.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 5.12.** What do you mean by intensity attenuation ?

**Answer**

1. Intensity attenuation refers to any reduction in the strength of a signal.
2. It is relatively compared with the attenuation of peak ground acceleration that is used for engineering design.
3. Knowledge of intensity attenuation is useful in calibrating hazard models against historical experience.
4. Intensity attenuation equation is given by :
$$I = 3.31 + 1.28 \, ML - 1.22 \ln R$$
   where         $I$ = Intensity
                 $ML$ = Local Magnitude
                 $R$ = Hypocentral distance
5. As radiant energy from a point light source travels through space, its amplitude is attenuated by the factor $1/d^2$, where $d$ is the distance that the light has travelled.
6. This means that a surface close to the light source receives higher incident intensity from the source than a distant surface.

**Que 5.13.** Write a short note on colour considerations.

**Answer**

1. Most graphics displays of realistic scenes are in colour. To incorporate colour, we need to write the intensity equation as a function of the colour properties of the light sources and object surfaces.
2. For an RGB description, each colour in a scene is expressed in terms of red, green, and blue components.
3. We then specify the RGB components of light source intensities and surface colours, and the illumination model calculates the RGB components of the reflected light.

4. One way to set surface colours is by specifying the reflectivity coefficients as three element vectors.

5. The diffuse reflection coefficient vector would then have RGB components ($K_{dR}$, $K_{dG}$, $K_{dB}$).

6. If we want an object to have a blue surface, we select a non-zero value in the range from 0 to 1 for the blue reflectivity component, $K_{dB}$, while the red and green reflectivity components are set to zero ($K_{dR} = K_{dG} = 0$).

7. Any non-zero red or green components in the incident light are absorbed, and only the blue component is reflected. The intensity calculation for this example reduces to the single expression

$$I_B = K_{aB}I_{aB} + \sum_{i=1}^{n} f_i(d)I_{lBi}[K_{dB}(N.L_i) + K_{sB}(N.H_i)^{n_s}]$$

8. Surfaces typically are illuminated with white light sources, and in general we can set surface colour so that the reflected light has non-zero values for all three RGB components.

9. Calculated intensity levels for each colour component can be used to adjust the corresponding electron gun in an RGB monitor.

---

**Que 5.14.** | **Write short note on transparency and shadows.**

**Answer**

**Transparency :**

1. A transparent surface produces both reflected and transmitted light.

2. The relative contribution of the transmitted light depends on the degree of transparency of the surface.

3. When a transparent surface is to be modeled, the intensity equations must be modified to include contributions from light passing through the surface.

4. Both diffuse and specular transmission can take place at the surfaces of a transparent object.

5. Realistic transparency effects are modeled by considering light refraction.

6. When light is incident upon a transparent surface, it is reflected and refracted, because the speed of light is different in different mediums so it changes the direction as shown in Fig. 5.14.1.

   where,

   Refraction direction = $T$

   Reflection direction = $R$

   Angle of refraction = $\theta_r$

   Angle of incidence $\theta_i$ = angle of reflection

   Distance travelled by light in transparent medium = $d$

   Index of refraction of the refracting material = $\eta_r$

Index of refraction of the incidents material = $\eta_i$
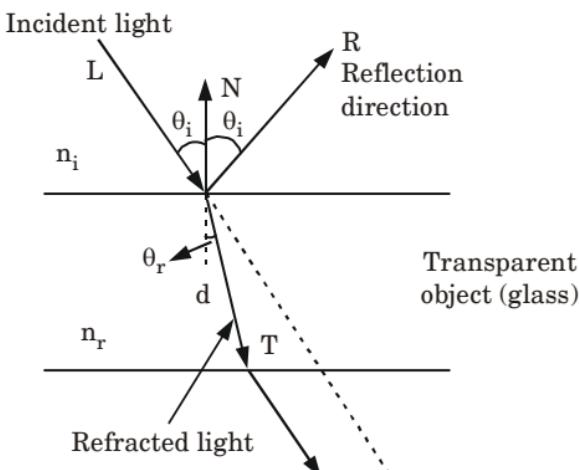
By Snell's law, we have

$$\sin \theta_r = \frac{\eta_i}{\eta_r} \sin \theta_i$$

From Snell's law and Fig. 5.14.1 we can obtain the unit transmission vector $T$ in the refraction direction $\theta_r$ as :

$$T = \left( \frac{\eta_i}{\eta_r} \cos \theta_i - \cos \theta_r \right) N - \frac{\eta_i}{\eta_r} L$$

where $N$ is the unit normal vector on the surface and $L$ is the unit vector in the direction of light source.



**Fig. 5.14.1.** Refraction light.

**Shadows :**

1. A shadowed object is one which is hidden from the light source.

2. It is possible to use hidden surface algorithms to locate the areas where light sources produce shadows.

3. In order to achieve this we have to repeat the hidden surface calculation using light source as the viewpoint.

4. This calculation divides the surfaces into shadowed and unshadowed groups.



**Fig. 5.14.2.** Objects modeled with shadow regions.

5. The surfaces that are visible from the light source are not in shadow; those that are not visible from the light source are in shadow.

6. Surfaces which are visible and which are also visible from the light source are shown with both the background illumination and the light-source illumination.

☺☺☺

# 1
UNIT

# Introduction and Line Generation (2 Marks Questions)

### 1.1. Define computer graphics.

**Ans.** Computer graphics is an art of drawing pictures on computer screens with the help of programming.

### 1.2. What are the advantages of computer graphics ?

**Ans. Advantages of computer graphics :**
   i. A high quality graphics displays of personal computer provide one of the most natural means of communication with a computer.
   ii. It provides tools for producing not only of concrete, real-world objects but also of abstract, synthetic objects.
   iii. The computer graphics also provides facility called update dynamics.

### 1.3. What are the application of computer graphics ?

AKTU 2018-19, Marks 02

**Ans. Applications of computer graphics :**
   i. User interfaces       ii. Plotting of graphics and chart
   iii. Office automation and desktop publishing
   iv. Simulation and animation
   v. Art and commerce

### 1.4. Explain the role of pixel and frame buffer in graphics devices.

AKTU 2015-16, Marks 02

**Ans. Role of pixel :** A pixel is a smallest size object or colour spot that can be displayed or addressed on a monitor. They represent the smallest controllable element of a picture on the screen.
**Role of frame buffer :** Picture definition is stored in a memory area called the refresh buffer or frame buffer. It holds the set of intensity values for all the screen points.

### 1.5. Describe random scan display and raster scan display.
### OR
**What do you understand by the raster scan display ?**

AKTU 2015-16, Marks 02

**Ans. Random scan display :** In random scan display, the beam is moved between the endpoints of the graphics primitive.
**Raster scan display :** Raster scan display is a method of displaying images on the CRT screen. In this, the horizontal and vertical

deflection signals are generated to move the beam all over the screen in a pattern.

**1.6. What do you understand by the video controller ?**

**Ans.** Video controller is one of the components of an interactive raster scan system. It is used to control the operation and intensity information of each pixel from frame buffer and displays them on screen.

**1.7. What are points and lines ?**

**Ans.** A point is a position is a plane. It has no size *i.e.*, no width, no length and no depth. A point is shown by dot.
A line is defined as a line of points that extends infinitely in two directions. It has one dimension *i.e.*, length.

**1.8. What are the criteria that should be satisfied by a good line drawing algorithm ?**

**Ans. Criteria for good line drawing algorithm are :**
   i. Lines should appear straight.
  ii. Lines should terminate accurately.
 iii. Lines should have constant density.
  iv. Lines should be drawn rapidly.

**1.9. Give the advantages of DDA algorithm.**

**Ans. Advantages of DDA algorithm :**
   i. It is the simplest algorithm and it does not require special skills for implementation.
  ii. It is a faster method for calculating pixel position.

**1.10. What are the advantages of B-spline curve over Bezier curve ?**

**Ans. Advantages of B-spline curve over Bezier curve :**
   i. The degree of B-spline polynomial can be set independently of the number of control points.
  ii. B-spline allows local control over the shape of a spline curve or surface.

**1.11. What are flat panel display devices ?**

**Ans.** Flat panel display is a display method that is designed to reduce the depth of the CRT display caused by the length of the tube.

**1.12. Describe the categories of flat panel displays.**

**Ans. Categories of flat panel display :**
   1. **Emissive displays :** These are the devices that convert electrical energy into light.
      Example : Plasma panels, thin-films etc.
   2. **Non-emissive displays :** They use optical effects to convert sunlight or light from some other source into graphics pattern.
      Example : Liquid Crystal Device (LCD).

**1.13. Write the Bresenham's algorithm of a line.**

**Ans.** **Bresenham's line drawing algorithm :**
1. Input the two lines end points and store the left end point $(x_0, y_0)$.
2. Load $(x_0, y_0)$ into frame buffer, *i.e.*, plot the first point.
3. Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for decision parameter as : $p_0 = 2\Delta y - \Delta x$
4. At each $x_k$ along the line, starting at $k = 0$, perform the following test :
   a. If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and
   $$p_{k+1} = p_k + 2\Delta y$$
   b. If $p_k > 0$, the next point to plot is $(x_k + 1, y_k + 1)$ and
   $$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$
5. Repeat step 4 $\Delta x$ times.

### 1.14. What is the persistence of phosphor ?

**AKTU 2015-16, Marks 02**

**Ans.** Persistence is defined as the time taken by the emitted light from the screen to decay to one-tenth of its original intensity. A phosphor with low persistence is useful for animation; a high persistence phosphor is useful for displaying highly complex, static pictures.

### 1.15. What is rasterization ?

**Ans.** Resterization is the task of taking an image described in a vector graphics format and converting it into raster image. The rasterized image may then be displayed on computer display, video display or stored in a bitmap file format.

### 1.16. Explain the term pixel, aspect ratio, resolution.

**Ans.** **Pixel :** A pixel is a smallest size object or colour spot that can be displayed or addressed on a monitor. They represent the smallest controllable element of a picture on the screen.
   **Aspect ratio :**
1. An aspect ratio is an attribute that describes the relationship between the width and height of an image.
2. Aspect ratio is expressed by the symbolic notation *i.e.*, $x : y$.
   **Resolution :**
1. Resolution is defined as the number of pixels on the horizontal axis and vertical axis.
2. The sharpness of the image on the display depends on the resolution and size of the monitor.

### 1.17. Define aspect ratio and types of retracing ?

**Ans.** **Aspect ratio :**
1. An aspect ratio is an attribute that describes the relationship between the width and height of an image.
2. Aspect ratio is expressed by the symbolic notation *i.e.*, $x : y$.
   **Types of retracing :** There are two types of retracing :
1. Vertical retracing       2. Horizontal retracing

☺☺☺

# 2 UNIT

# Transformation (2 Marks Questions)

### 2.1. Define transformation.

**Ans.** Transformation is the fundamental ability to simulate the manipulation of objects in space, *i.e.*, it is used to position, reshape and animate objects, lights and the virtual camera.

### 2.2. List out the basic operations performed by geometric transformation.

**OR**

**Explain other transformation that can be applied on 2D objects ?**

AKTU 2018-19, Marks 02

**Ans.** **Basic operations performed by geometric transformation are :**

i. Translation
ii. Rotation
iii. Scaling

### 2.3. What do you understand by reflection ?

**Ans.** A reflection is a transformation that produces the mirror image of an object.

### 2.4. What do you mean by shearing ?

**Ans.** Shearing is a transformation that slants (or lean) the shape of an object in a particular direction *i.e.*, vertical or horizontal.

### 2.5. Define homogeneous coordinates.

**Ans.** Homogeneous coordinates consists of appending a 1 to the *x* and *y* coordinates.

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In order to translate as a matrix multiplication operation, we use $3 \times 3$ matrices and transform the points to become $3 \times 1$ matrices.

### 2.6. Define windowing and when it is used.

**Ans.** Windowing is a process of transforming coordinates from one place to another. It is used when scaling transformation is done.

### 2.7. Define clipping.

**Ans.** Clipping is the method to cut a specific object. The object may be point, area, line or text. It is the process which divides each element of the picture into its visible and invisible portions.

### 2.8. What is line clipping ?

**Ans.** Line clipping is a method to cut a specific line. It removes portion of line segment outside window. A line is completely visible if both of its end points are in the window.

### 2.9. Define point clipping.

**Ans.** Points are said to be interior to the clipping window if,

$$x_{w\min} \leq x \leq x_{w\max} \quad \text{and} \quad y_{w\min} \leq y \leq y_{w\max}$$

The equal sign indicates that the points on the window boundary are included within the window.

### 2.10. What does the word "Filling Polygon" mean ?

**Ans.** "Filling Polygon" is the process of colouring in a fixed area or region. There are various algorithms to fill the area :
   i. Boundary-fill algorithm
   ii. Flood-fill algorithm
   iii. Scanline algorithm

### 2.11. Write a note on text clipping.

**Ans.** Text clipping is a method in which either a part of character that is not inside the window boundary or a character outside a window boundary is clipped.

### 2.12. Write a note on curve clipping.

**Ans.** Curve clipping involves non-linear equations and require more complex processing than for objects with linear boundaries. The bounding rectangle for a circle or other curved object can be used first to test for overlap with a rectangular clip window.

### 2.13. Justify the composite transformation.

> **AKTU 2015-16, Marks 02**

**Ans.**
1. A composite transformation is two or more transformations performed one after the other.
2. If a transformation of the plane $T_1$ is followed by a second plane transformation $T_2$, then the result itself may be represented by a single transformation $T$ which is the composition of $T_1$ and $T_2$ taken in that order.
3. This is written as $T = T_1.T_2$.
4. Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.

**2.14. Give window to viewport transformation matrix.**

AKTU 2015-16, Marks 02

**Ans.**

$$\left[\begin{array}{ccc} S_x = \dfrac{V_{x\max} - V_{x\min}}{W_{x\max} - W_{x\min}} & 0 & 0 \\[3ex] 0 & S_x = \dfrac{V_{y\max} - V_{y\min}}{W_{y\max} - W_{y\min}} & 0 \\[3ex] 0 & 0 & 1 \end{array}\right]$$

**2.15. List out the types of polygon clipping algorithm.**

**Ans. Types of polygon clipping algorithm are :**
i. Sutherland-Hodgeman polygon clipping
ii. Weiler-Atherton polygon clipping
iii. Curve clipping

**2.16. Define translation and rotation.**

**Ans. Translation :** Translation is a process of changing the position of an object in a straight line path from one coordinate location to another.

**Rotation :** Two dimensional rotation is applied to an object by rotating it along a circular path in the *xy* plane.

**2.17. Distinguish between window port and view port.**

**Ans.**

| S. No. | Window port | View port |
|--------|-------------|-----------|
| 1. | World coordinates area selected for display is called a window port. | An area on a display device which is mapped is called a view port. |
| 2. | Window port defines what is to be viewed. | View port defines where it is to be displayed. |

**2.18. How many clippers are used by Sutherland Hodgeman for polygon clipping ?**

AKTU 2018-19, Marks 02

**Ans.** Four clippers are used by Sutherland Hodgeman for polygon clipping.

☺☺☺

# 3
UNIT

# Three Dimensional
# (2 Marks Questions)

### 3.1. What do you mean by geometric primitives ?

**Ans.** Geometric primitives are the basic geometrical shapes used to construct computer graphic scenes and the resulting final images.

### 3.2. Name the commonly used 3D geometric primitives.

**Ans.** **Commonly used 3D geometric primitives are :**
   i. Points
   ii. Line
   iii. Line segments
   iv. Planes
   v. Circles
   vi. Ellipses
   vii. Triangles

### 3.3. Write the rotation matrix about *x*, *y* and *z*-axis in 3D.

AKTU 2015-16, Marks 02

**Ans.**

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 3.4. How can we determine the projected view of an object ?

**Ans.** The projected view of an object is determined by calculating the intersection of the projection lines with the view plane.

### 3.5. What is projection ? Give its types.

**Ans.** Projection is the transformation of a three-dimensional (3D) area into a two-dimensional (2D) area.

**Types of projection are :**
i. Perspective projection
ii. Parallel projection

### 3.6. Discuss the classification of parallel projection.

**Ans.** **Parallel projections are classified into two ways :**
  i. **Orthographic :** In this, projections are perpendicular to view plane.
  ii. **Oblique :** In this, projections are not perpendicular to view plane.

### 3.7. Give the common subcategories of orthographic projection.

**Ans.** **Subcategories of orthographic projection are :**
  i. Isometric
  ii. Diametric
  iii. Trimetric

### 3.8. Give the common subcategories of oblique projection.

**Ans.** **Subcategories of oblique projection are :**
  i. Cabinet
  ii. Cavalier

### 3.9. Define perspective projection.

**Ans.** In perspective projection, object positions are transformed to the view plane along lines that converge to a point called the project reference point.

### 3.10. Write down the anomalies of perspective projection.

**Ans.** **Anomalies of perspective projection are :**
  i. Perspective foreshortening
  ii. Vanishing point
  iii. View confusion
  iv. Topological distortion

### 3.11. Write down the types of perspective projection.

**Ans.** **Types of perspective projection are :**
  i. One-point perspective projection
  ii. Two-point perspective projection
  iii. Three-point perspective projection

### 3.12. Define 3D clipping.

**Ans.** Clipping in three dimensions can be accomplished using extensions of two-dimensional clipping methods. In 3D clipping, we clip the objects against the boundary planes of the view volume.

### 3.12. Define vanishing point.

**Ans.** A vanishing point is an abstract point on the image plane where 2D projections (or drawings) of a set of parallel lines in 3D space appear to converge.

**3.13. What is tilting transformation ? Does the order of performing the rotation matter ?**

**Ans.** Tilting is a transformation, where the object first rotate about $x$-axis and then about $y$-axis. We can find the required tranformation by composing two rotation matrices, *i.e.*,

$$R_{\theta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_{\theta_y} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiply $R_{\theta_y}$ and $R_{\theta_x}$, we get

$$R_{\theta_y} \cdot R_{\theta_x} = \begin{bmatrix} \cos\theta_y & \sin\theta_y\sin\theta_x & \sin\theta_y\cos\theta_x & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ -\sin\theta_y & \cos\theta_y\sin\theta_x & \cos\theta_y\cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, we multiply $R_{\theta_x}$ and $R_{\theta_y}$, we get

$$R_{\theta_x} \cdot R_{\theta_y} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ \sin\theta_x\sin\theta_y & \cos\theta_x & -\sin\theta_x\cos\theta_y & 0 \\ -\cos\theta_x\sin\theta_y & \sin\theta_x & \cos\theta_x\cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, $R_{\theta_x} \cdot R_{\theta_y}$ is not the same matrix, as $R_{\theta_y} \cdot R_{\theta_x}$. Thus, the order of rotation matrix is important.

☺☺☺

# 4
**UNIT**

# Curves and Surfaces (2 Marks Questions)

### 4.1. Define quadric surface.

**Ans.** Quadric surface is a generalization of conic sections. It is hyper surface (of dimension $D$) in a $(D + 1)$ dimensional space.

### 4.2. Name the various types of quadric surfaces.

**Ans. Various types of quadric surface are :**
  i. Ellipsoid
  ii. Sphere
  iii. Torus
  iv. Blobby objects

### 4.3. Define blobby objects.

**Ans.** Blobby objects do not maintain a fixed shape and change their surface characteristics in certain motions or when they come in contact with other objects.

### 4.4. Define spline curves.

**Ans.** A spline curve is a flexible strip used to produce a smooth curve through a designated set of points. A spline curve is any composite curve with polynomial section satisfying some specified continuity conditions.

### 4.5. What is B-spline ?

**Ans.** B-spline is a basis spline function which contains a set of control points.

### 4.6. How many parametric continuity conditions are in a spline representation ?

**Ans. Parametric continuity conditions in a spline representation are :**
  i. Zero-order parametric continuity
  ii. First-order parametric continuity
  iii. Second-order parametric continuity

### 4.7. Write down the properties of Bezier curves.

**Ans.** **Properties of Bezier curves are :**
   i. It always passes through the first and last point.
   ii. It lies within the convex hull of the control points.
   iii. The sum of all Bezier blending function is equal to one.

### 4.8. Write the properties of B-spline curves.

**Ans.** **Properties of B-spline curves are :**
   i. The polynomial curve has degree $d-1$ and $C^{d-2}$ continuity over the range of 4.
   ii. For $n+1$ control points, the curve is described with $n+1$ blending functions.

### 4.9. What are the advantages of B-spline curve over Bezier curve ?

**Ans.** **Advantages of B-spline curve over Bezier curve :**
   i. The degree of B-spline polynomial can be set independently of the number of control points.
   ii. B-spline allows local control over the shape of a spline curve or surface.

### 4.10. Define blobby objects and types of coherence.

**Ans.** **Blobby objects :** Blobby objects do not maintain a fixed shape and change their surface characteristics in certain motions or when they come in contact with other objects.
   **Types of coherence :** There are two type of coherence :
   1. Spatial coherence
   2. Temporal coherence

☺☺☺

# 5
**UNIT**

# Hidden Lines and Surfaces
# (2 Marks Questions)

**5.1. What is determined by the hidden line or hidden surface ?**

**Ans.** Hidden line or hidden surface determines the lines, edges, surfaces or volumes that are visible or invisible to an observer located at a specific point in space.

**5.2. Mention the two approaches for hidden surface elimination or visible surface detection.**

**Ans.** Two approaches for hidden surface elimination are :
   i. Object space method       ii. Image space method

**5.3. Give the advantages of z-buffer algorithm.**

**Ans.** Advantages of z-buffer algorithm are :
   i. It is easy to implement
   ii. It can be implemented in hardware to overcome the speed problem.
   iii. The total number of polygons in pictures can be large.

**5.4. What are the various back-face detection algorithms ?**

**Ans.** Various back-face detection algorithms are :
   i. Back-face detection methods       ii. Depth buffer method
   iii. A-buffer method                 iv. Scan line method

**5.5. Mention the name of the basic illumination models.**

**Ans.** Basic illumination models are
   i. Ambient light                 ii. Diffuse reflection
   iii. Specular reflection         iv. Phong model
   v. Combined approach             vi. Warn model
   vii. Intensity attenuation

**5.6. Describe diffuse illumination.**

**Ans.** An object may be illuminated by light which does not come from any particular source but which comes from all directions. When such illumination is uniform from all directions, the illumination is called diffuse illumination.

**5.7. What do you understand by coefficient of reflection ?**

**Ans.** Coefficient of reflection is the ratio of the light reflected from the surface to the total incoming light to the surface.

**5.8. What is the specular reflection ?**

> **AKTU 2015-16, Marks 02**

**Ans.** Specular reflection is the phenomenon of reflection of incident light in a concentrated region around the specular reflection angle.

**5.9. Write an equation for the phong reflection model.**

**Ans.** $$I_{\text{spec}} = W(\theta)\, I_l \cos^{ns} \phi$$

where $I_l$ is the intensity of the light source, $n_s$ is specular reflection parameter and $\phi$ is the viewing angle relative to specular reflection direction $R$.

**5.10. Give the combine diffuse and specular reflection from any point on the illuminated surface for a single point light source.**

**Ans.** $$I = I_{\text{diff}} + I_{\text{spec}} = k_a I_a + k_d I_l\,(N.L) + k_s I_l\,(N.H)^{ns}$$

where,

$k_d$ is diffused-reflection coefficient

$k_a$ is ambient-reflection coefficient

$I_a$ is ambient light intensity for each surface

$I_l$ is intensity of point light source.

$N$ is unit normal vector to a surface

$L$ is unit direction vector to the point light source

$H$ is halfway vector

**5.11. On which factor the coefficient of transparency depends ?**

**Ans.** The coefficient of transparency depends on the thickness of the object because the transmission of light depends exponentially on the distance which the light ray must travel within the object.

**5.12. Define shadows.**

**Ans.** A shadow is one which is hidden from the light source. It is possible to use hidden surface algorithms to locate the areas where light source produces shadows.

**5.13. Define filled area primitives. What are the common methods used ?**

**Ans.** Filling is the process of 'coloring in' a fixed area or region.
**Common methods used are :**
1. Boundary-fill algorithm     2. Flood-fill algorithm

**5.14. What do you understand by match band effect and transparency ?** | AKTU 2018-19, Marks 02

**Ans. Mach band effect :** The linear intensity interpolation can result bright or dark intensity streaks to appear on the surface. These bright or dark intensity streaks, are called mach bands. The mach band effect can be reduced by breaking the surface into a greater number of smaller polygons.

**Transparency :**
1. Transparency is a phenomenon in which the object in the way of light ray does not obstruct light.
2. It means that if the light source is behind the object and the object is transparent, then the viewer can see the light through the object.

☺☺☺

# B.Tech.
# (SEM. V) ODD SEMESTER THEORY
# EXAMINATION, 2012-13
# COMPUTER GRAPHICS

**Time : 3 Hours**                                    **Max. Marks : 100**

**Note :** Attempt all **four** questions. Attempt any **two** parts in each question.

1. a. **Explain raster scan display device and random scan display device.**

   b. **Develop and write Bresenham's algorithm of line. Also predict the pixels on the line from (2, 2) to (12, 10) using this algorithm.**

   c. **Explain the parallel version of line algorithm by two methods.**

2. a. **Establish the reflection transformation matrix about the line $y = mx + b$.**

   b. **Write Liang and Barsky line clipping algorithm. Apply it for calculating the saved portion of line from (2,7) to (8,12) in a window.**
   **($x_{wmin} = y_{wmin} = 5$ and $x_{wmax} = y_{wmax} = 10$).**

   c. **Write Sutherland-Hodgeman polygon clipping algorithm. Explain the modification given by Weiler and Atherton for concave polygon.**

3. a. **Derive rotation about $x$-axis, $y$-axis and $z$-axis matrices in 3D. Also, prove that for any rotation matrix $R^{-1}(\theta) = R(-\theta)$.**

   b. **Derive oblique parallel projection and perspective projection matrices.**

   c. **Establish and write Cohen and Sutherland 3D line clipping algorithm.**

4. a. **Write at least four properties of Bezier curves. Calculate and also, roughly trace the Bezier curve for three control points (1,1), (2,2) and (3,1).**

    **b. Explain specular reflection and Phong model.**

    **c. Write back-face detection algorithm of hidden lines.**

☺☺☺

# SOLUTION OF PAPER (2012-13)

**Note :** Attempt all **four** questions. Attempt any **two** parts in each question.

**1. a. Explain raster scan display device and random scan display device.**

**Ans.** **Raster scan display :**
1. In raster scan displays, screen is scanned in horizontal and vertical direction and the information is stored in a buffer called frame buffer.
2. The frame buffer is used to store intensity values of all screen points.
3. It is suitable for displaying realistic scenes containing either complex shades or colour patterns.
4. Simple black and white display require only one bit per pixel while colour display systems require multiple bits per pixel.
5. Refreshing on raster scan displays is carried out at the rate of 60 to 80 frames per second.

**Random scan display :**
1. In random scan display, the definition of picture is stored as a collection of line of commands in an area of memory called refresh buffer or display program.
2. Random scan display draw a picture on line at a time and for this reason are also referred to as vector displays.
3. It is basically designed for line drawing and not suitable for complex natural scenes.
4. It refreshes at a rate of 30 to 60 frames per second.

**b. Develop and write Bresenham's algorithm of line. Also predict the pixels on the line from (2, 2) to (12, 10) using this algorithm.**

**Ans.** **Bresenham's line drawing algorithm :**
1. Input the two lines endpoints and store the left endpoint $(x_0, y_0)$.
2. Load $(x_0, y_0)$ into frame buffer, *i.e.*, plot the first point.
3. Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for decision parameter as :
$$p_0 = 2\Delta y - \Delta x$$
4. At each $x_k$ along the line, starting at $k = 0$, perform the following test :
   a. If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and
   $$p_{k+1} = p_k + 2\Delta y$$
   b. If $p_k > 0$, the next point to plot is $(x_k + 1, y_k + 1)$ and
   $$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$
5. Repeat step 4  $\Delta x$ times.

**Numerical :**

$$x_1 = 2, y_1 = 2, x_2 = 12, y_2 = 10$$
$$\Delta x = x_2 - x_1 = 12 - 2 = 10$$
$$\Delta y = y_2 - y_1 = 10 - 2 = 8$$

Slope of the line $\quad m = (y_2 - y_1)/(x_2 - x_1) = (10 - 2)/(12 - 2) = 0.8$

The initial decision parameter $p_0 = 2\Delta y - \Delta x$

$$p_0 = 2 \times 8 - 10 = 6$$

The increments for calculating successive decision parameters are $2\Delta y - 2\Delta x = 16 - 20 = -4$ and $2\Delta y = 16$, plot the initial point $(x_0, y_0) = (2, 2)$, successive are :

| Step | $p_k$ | Pixel |
|------|------|---------|
| 0 | 6 | (3, 3) |
| 1 | 2 | (4, 4) |
| 2 | – 2 | (5, 4) |
| 3 | 14 | (6, 5) |
| 4 | 10 | (7, 6) |
| 5 | 6 | (8, 7) |
| 6 | 2 | (9, 8) |
| 7 | – 2 | (10, 8) |
| 8 | 14 | (11, 9) |
| 9 | 10 | (12, 10) |



**Fig. 1.**

**c. Explain the parallel version of line algorithm by two methods.**

**Ans.** In the parallel version of line algorithm, pixel positions are calculated along a line path simultaneously by partitioning the computations

among the various processors available. Methods for parallel version of line algorithm are :

**1. Bresenham's line algorithm :**

a. Given $n_p$ processors, a parallel Bresenham's line algorithm can be used by subdividing the line path into $n_p$ partitions and simultaneously generating line segments in each of the sub-intervals.

b. For a line with slope $0 < m < 1$ and left endpoint coordinate $(x_0, y_0)$, the line is partitioned along the positive $x$-direction.

c. The distance between beginning $x$ positions of adjacent partitions is calculated as :

$$\Delta x_p = \frac{\Delta x + n_p - 1}{n_p}$$

where $\Delta x$ is the width of the line and the value for partition width $\Delta x_p$ is computed using integer division.

d. The starting $x$ coordinate is calculated for the $k$th partition as :

$$x_k = x_0 + k\Delta x_p$$

e. At the $k$th partition, the starting $y$ coordinate is :

$$y_k = y_0 + \text{round } (k\Delta y_p)$$

f. The initial decision parameter for Bresenham's algorithm at the start of the $k$th sub interval is :

$$p_k = (k\Delta x_p)(2\Delta y) - \text{round } (k\Delta y_p)(2\Delta x) + 2\Delta y - \Delta x$$

**2. In raster scan system :**

a. In raster systems we assign each processor to a particular group of screen pixel.

b. For lines with a positive slope greater than 1, we reverse the value of $x$ and $y$. *i.e.*, we sample at unit $y$ intervals ($\Delta y = 1$) and calculate each succeeding $x$ value as

$$x_{k+1} = x_k + \frac{1}{m}$$

c. Lines are to be processed from the left endpoint to the right endpoint. If this processing is reversed, so that the starting endpoint is at the right, then either we have $\Delta x = -1$ and

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

or (when the slope is greater than 1) we have $\Delta y = -1$ with

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

d. Perpendicular distance $d$ from the line in to pixel as shown in Fig. 2 with coordinates $(x, y)$ is obtained with the calculation

$$d = Ax + By + C$$
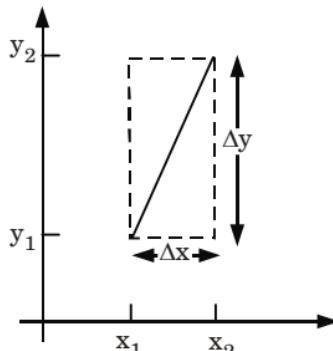
where,                    $$A = \frac{-\Delta y}{\text{linelength}}$$

$$B = \frac{\Delta x}{\text{linelength}}$$

$$C = \frac{x_0 \Delta y - y_0 \Delta x}{\text{linelength}}$$
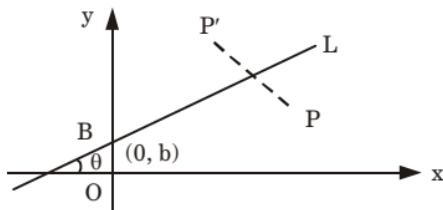
$$\text{linelength} = \sqrt{\Delta x^2 + \Delta y^2}$$



**Fig. 2.**

**2. a. Establish the reflection transformation matrix about the line $y = mx + b$.**

**Ans.** Let line $L(y = mx + b)$ have a $y$-intercept $(0, b)$ and an angle $\theta$ with respect to $x$-axis. The steps involved in reflection transformation are as follows :

1. Translate the intersection point $B$ to the origin.
2. Rotate by $-\theta°$ so that the line $L$ aligns with the $x$-axis.
3. Mirror reflect about the $x$-axis.
4. Rotate back by $\theta°$.
5. Translate $B$ back to $(0, b)$.



**Fig. 3.** Reflection of point $P$ about line $L$.

Therefore, we have

$$[M_L = T_V . R_\theta . M_x . R_{-\theta} . T_{-V}]$$

Where $\qquad V = bJ$

$$T_V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{-V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{-\theta} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Slope,    $m = \tan\theta$

Therefore,  $\sin\theta = \dfrac{m}{\sqrt{m^2+1}}$

$\cos\theta = \dfrac{1}{\sqrt{m^2+1}}$

After multiplication of all basic matrix, we get

$$M_{L} = \begin{bmatrix} \dfrac{1-m^2}{m^2+1} & \dfrac{2m}{m^2+1} & \dfrac{-2bm}{m^2+1} \\[2ex] \dfrac{2m}{m^2+1} & \dfrac{m^2-1}{m^2+1} & \dfrac{2b}{m^2+1} \\[2ex] 0 & 0 & 1 \end{bmatrix}$$

**b. Write Liang and Barsky line clipping algorithm. Apply it for calculating the saved portion of line from (2,7) to (8,12) in a window.**

$(x_{w\min} = y_{w\min} = 5$ and $x_{w\max} = y_{w\max} = 10).$

**Ans.** **The Liang-Barsky algorithm :**

1. Read two endpoint of the line say $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$.
2. Read two corners (left-top and right-bottom) of the window, say $(x_{w\min}, y_{w\max}, x_{w\min}, y_{w\min})$.
3. Calculate the values of parameters $p_i$ and $q_i$ for $i = 1, 2, 3, 4$ such that

   $p_1 = -\Delta x$         $q_1 = x_1 - x_{w\min}$
   $p_2 = \Delta x$          $q_2 = x_{w\max} - x_1$
   $p_3 = -\Delta y$         $q_3 = y_1 - y_{w\min}$
   $p_4 = \Delta y$          $q_4 = y_{w\max} - y_1$
4. If $p_i = 0$, then

{   The line is parallel to $i^{th}$ boundary.

Now, if $q_i < 0$ then

{  line is completely outside the boundary, hence discard the line segment and goto stop.

}

else

{  Check whether the line is horizontal or vertical and accordingly check the line endpoint with corresponding boundaries. If line's endpoints lie within the bounded area then use them to draw line otherwise use boundary coordinates to draw line. Go to stop.

}

}

5.  Initialize values for $t_1$ and $t_2$ as $t_1 = 0$ and $t_2 = 1$.

6.  Calculate values for $q_i/p_i$ for $i = 1, 2, 3, 4$.

7.  Select values of $q_i/p_i$ where $p_i < 0$ and assign maximum out of them as $t_1$.

8.  Select values of $q_i/p_i$ where $p_i > 0$ and assign minimum out of them as $t_2$.

9.  If $(t_1 < t_2)$

{  Calculate the endpoints of the clipped line as follows :

$xx_1 = x_1 + t_1 \Delta x$

$xx_2 = x_1 + t_2 \Delta x$

$yy_1 = y_1 + t_1 \Delta y$

$yy_2 = y_1 + t_2 \Delta y$

Draw line $(xx_1, yy_1, xx_2, yy_2)$

}

10. Stop.

**Numerical :**

$$P_1 = (2, 7)$$
$$P_2 = (8, 12)$$
$$(x_{w\min}, y_{w\min}) = (5, 5)$$
$$(x_{w\min}, y_{w\max}) = (5, 10)$$
$$(x_{w\max}, y_{w\min}) = (10, 5)$$
$$(x_{w\max}, y_{w\min}) = (10, 10)$$
$$dx = 8 - 2 = 6$$
$$dy = 12 - 7 = 5$$

$p_1 = -dx = -6$    $q_1 = x_1 - x_{w\min} = 2 - 5 = -3$

$p_2 = dx = 6$    $q_2 = x_{w\max} - x_1 = 10 - 2 = 8$

$p_3 = -dy = -5$    $q_3 = y_1 - y_{w\min} = 7 - 5 = 2$

$p_4 = dy = 5$    $q_4 = y_{w\max} - y_1 = 10 - 7 = 3$

$$\frac{q_1}{p_1} = \frac{-3}{-6} = \frac{1}{2}$$

$$\frac{q_2}{p_2} = \frac{8}{6} = \frac{4}{3}$$

$$\frac{q_3}{p_3} = \frac{2}{-5} = -\frac{2}{5}$$

$$\frac{q_4}{p_4} = \frac{3}{5} = \frac{3}{5}$$

$$\text{or } (p_i < 0) \; t_1 = \text{Max} \left( \frac{1}{2}, -\frac{2}{5}, 0 \right) = \frac{1}{2}$$

$$\text{or } (p_i > 0) \; t_2 = \text{Min} \left( \frac{4}{3}, \frac{3}{5}, 1 \right) = \frac{3}{5}$$

$$t_1 < t_2$$
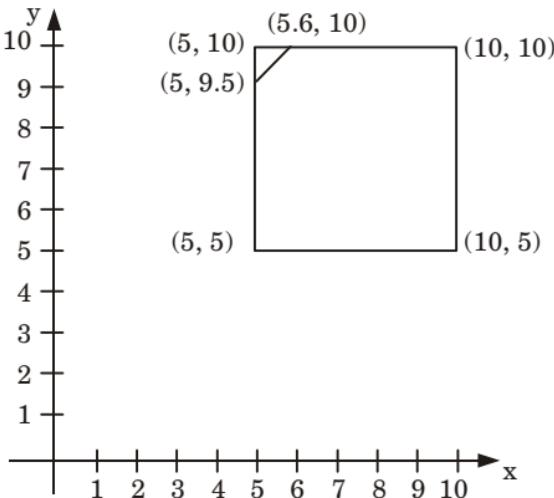
$$\therefore \qquad xx_1 = x_1 + t_1 dx = 2 + \frac{1}{2} \; (6) = 5$$

$$xx_2 = x_1 + t_2 dx = 2 + \frac{3}{5} \; (6) = 5.6$$

$$yy_1 = y_1 + t_1 dy = 7 + \frac{1}{2} \; (5) = 9.5$$

$$yy_2 = y_1 + t_2 dy = 7 + \frac{3}{5} \; (5) = 10$$

Coordinates of clipped lines will be (5, 9.5) and (5.6, 10).



**Fig. 4.**

**c. Write Sutherland-Hodgeman polygon clipping algorithm. Explain the modification given by Weiler and Atherton for concave polygon.**

**Ans.** **Sutherland-Hodgeman polygon clipping algorithm :**
1. Read coordinates of all vertices of the polygon.
2. Read coordinates of the clipping window.
3. Consider the left edge of the window.

4. Compare the vertices of each edge of the polygon, individually with the clipping plane.

5. Save the resulting intersections and vertices in the new list of vertices according to four possible cases between the edge and the clipping boundary.

6. Repeat the step 4 and 5 for remaining edges of the clipping window. Each time the resultant list of vertices is successively passed, move the process to the next edge of the clipping window.

7. Stop.

**Weiler-Atherton polygon clipping :**

1. In this algorithm, the vertex processing procedure is modified so that the concave polygons are displayed correctly.

2. This algorithm depends on identifying surfaces as shown in Fig. 5.



Clipping a concave polygon by applying Weiler-Atherton algorithm

Output (two separate polygon areas)

**Fig. 5.**

3. There are two directions (clockwise or anticlockwise) that exist to process the polygon vertices.

4. For clockwise processing of polygon vertices, we use the following rules :

   a. For an outside to inside pair of vertices, follow the polygon boundary.

   b. For an inside to outside pair of vertices, follow the window boundary in a clockwise direction.

**3. a. Derive rotation about $x$-axis, $y$-axis and $z$-axis matrices in 3D. Also, prove that for any rotation matrix $R^{-1}(\theta) = R(-\theta)$.**

**Ans.**

**i. Rotation about $x$-axis (*i.e.* axis of rotation is $x$) :**

$$x' = x$$
$$y' = r \cos (\alpha + \theta)$$
$$y' = r \cos \alpha \cos \theta - r \sin \alpha \sin \theta \qquad ...(1)$$

Since $\quad r \cos \alpha = y$ and $r \sin \alpha = z$

Put the value in eq. (1) we get,

$$y' = y \cos \theta - z \sin \theta$$
$$z' = r \sin (\alpha + \theta)$$

or, $\qquad\qquad = r \sin \alpha \cos \theta + r \cos \alpha \sin \theta$

$\qquad\qquad = z \cos \theta + y \sin \theta$
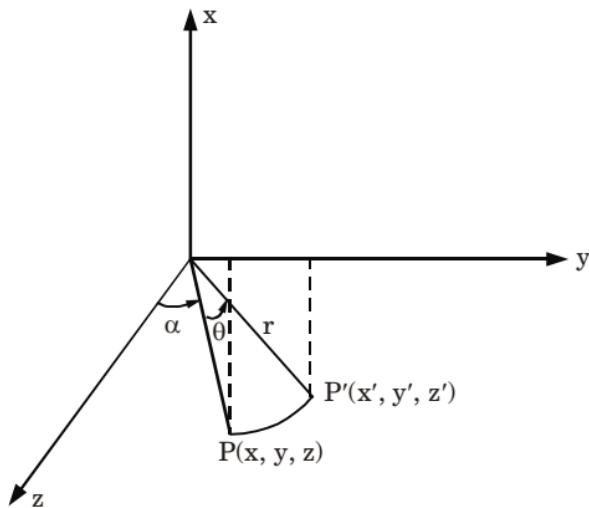
or, $R_{\theta,i} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$



**Fig. 6.** Rotation when axis of rotation is *x*-axis.
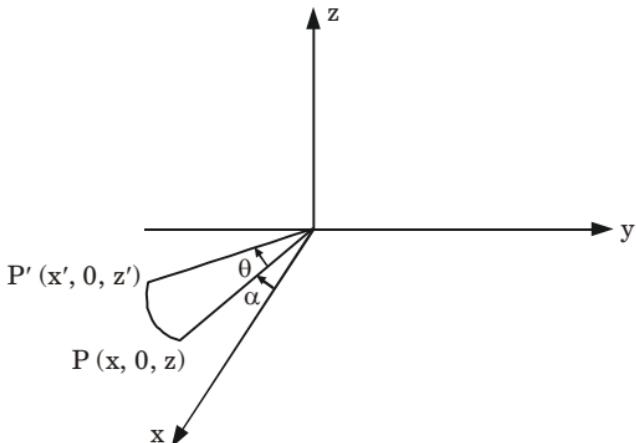
The points after rotation can be calculated by the following equation :

$$p'(x', y', z') = R_{\theta,i} P(x, y, z)$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**ii. Rotation about *y*-axis (*i.e.*, *y*-axis is axis of rotation) :**



**Fig.7.**

$$x' = x \cos\theta + z \sin\theta$$
$$y' = y$$

$$z' = -x \sin \theta + z \cos \theta$$
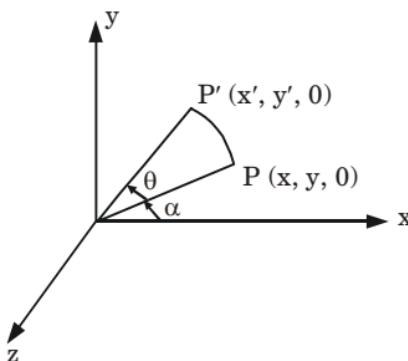
or,
$$R_{\theta, i} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**iii. Rotation about the z-axis (i.e., z-axis is axis of rotation)**



**Fig. 8.**

$$x' = x \cos \theta - y \sin \theta$$

$$y' = z \cos \theta + y \sin \theta$$

$$z' = z$$

or,
$$R_{\theta, k} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Numerical :**

Let
$$R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Put $\theta = (-\theta)$, we get

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) & 0 \\ 0 & \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since $\cos(-\theta) = \cos(\theta)$, $\sin(-\theta) = -\sin\theta$, we have

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now,

$$R(-\theta) * R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

We have,

$$R(-\theta) \cdot R(\theta) = I$$

Multiply $R^{-1}(\theta)$ both sides,

$$R(-\theta).R(\theta).R^{-1}(\theta) = I.R^{-1}(\theta)$$
$$R(-\theta) = R^{-1}(\theta)$$

Hence proved.

**b. Derive oblique parallel projection and perspective projection matrices.**

**Ans.** **Derivation of oblique parallel projection :**

1. An oblique projection is obtained by projecting points along parallel lines that are not perpendicular to the projection plane.
2. An oblique projection of a point $P(x, y, z)$ on the $z = 0$ plane is shown in Fig. 9.
3. $P_{ob}(x_p, y_p)$ is oblique projection of the point $P$ and $P_{or}(x, y)$ is the orthographic projection of $P$ on the $z = 0$ plane.
4. Let $L$ be the length of the line joining points $P_{ob}(x_p, y_p)$ and $P_{or}(x, y)$, $\theta$ be the angle made by the line $L$ with $x$-axis.
5. We can express the projection coordinates in terms of $x, y, L$ and $\theta$ as
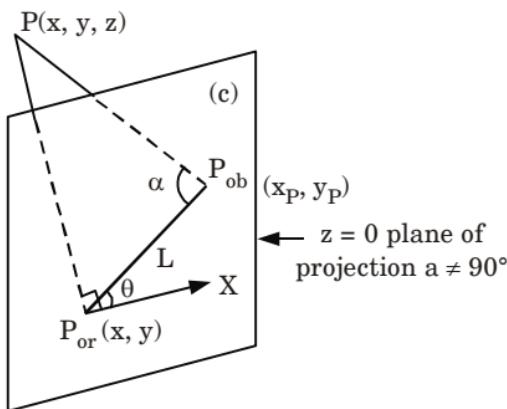
$$x_p = x + L\cos\theta$$
$$y_p = y + L\sin\theta$$

6. Length $L$ depends on the angle $\alpha$ and the $Z$ coordinate of the point to be projected

$$\tan \alpha = \frac{Z}{L} \Rightarrow L = \frac{Z}{\tan \alpha} = ZL_1 \quad \left( \text{where } L_1 = \frac{1}{\tan \alpha} \right)$$

when $\quad Z = 1, L = L_1$



**Fig. 9.**

7. Now, we can write the oblique projection as

$$x_p = x + Z(L_1 \cos \theta)$$
$$y_p = x + Z(L_1 \sin \theta)$$

8. Considering $Z_p = 0$, we get

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L_1 \cos \theta & 0 \\ 0 & 1 & L_1 \sin \theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Derivation of perspective projection matrix :**

1. Let us consider the center of projection is at $(x_c, y_c, z_c)$ and the point on object is $(x_1, y_1, z_1)$, then the parametric equation for the line containing these points can be given as

$$x_2 = x_c + (x_1 - x_c)u$$
$$y_2 = y_c + (y_1 - y_c)u$$
$$z_2 = z_c + (z_1 - z_c)u$$

where $u$ is the parameter

2. For projected point $z_2$ is 0, therefore, the third equation can be written as

$$0 = z_c + (z_1 - z_c)u$$

$$\therefore \qquad u = - \frac{z_c}{(z_1 - z_c)}$$

3. Put this value of $u$ in first two equations, we get

$$x_2 = x_c - \frac{z_c}{(z_1 - z_c)} (x_1 - x_c)$$

$$= \frac{x_c z_1 - x_c z_c - x_1 z_c + x_c z_c}{(z_1 - z_c)}$$

$$\therefore \qquad x_2 = \frac{x_c z_1 - x_1 z_c}{(z_1 - z_c)}$$

and

$$y_2 = y_c - \frac{z_c}{(z_1 - z_c)} (y_1 - y_c)$$

$$= \frac{y_c z_1 - y_c z_c - y_1 z_c + y_c z_c}{(z_1 - z_c)}$$

$$y_2 = \frac{y_c z_1 - y_1 z_c}{(z_1 - z_c)}$$

4. So, the homogeneous matrix can be written as

$$[x_2 \quad y_2 \quad z_2 \quad 1] = [x_1 \quad y_1 \quad z_1 \quad 1] \begin{bmatrix} -z_c & 0 & 0 & 0 \\ 0 & -z_c & 0 & 0 \\ x_c & y_c & 0 & 1 \\ 0 & 0 & 0 & -z_c \end{bmatrix}$$

**c. Establish and write Cohen and Sutherland 3D line clipping algorithm.**

**Ans.** **Cohen-Sutherland line clipping algorithm :**

1. In this algorithm, a rectangular window is considered whose coordinates are $(xw_{min}, yw_{min}, xw_{max}, yw_{max})$ as shown in Fig. 10.



**Fig. 10.**

2. Every line end point is assigned a 4-digit binary code (region code) that identifies the location of the point corresponding to the boundaries of the clipping window.

3. The procedure for calculating the end point $p(x, y)$ code is as follow :

**a. For bit 1**
If $(y - yw_{max}) < 0$ then
code = 0;
else
code = 1;

    **b. For bit 2**
    If $(yw_{min} - y) < 0$ then
    code = 0;
    else
    code = 1;
    **c. For bit 3**
    if$(x - xw_{max}) < 0$ then
    code = 0;
    else
    code = 1;
    **d. For bit 4**
    if$(xw_{min} - x) < 0$ then
    code = 0;
    else
    code = 1;

4. If the region code of both the end points of a line segment is 0000, the line is completely visible as it completely lies within the window.

5. If the region code of both the end points of the line segment is not 0000 then we calculate AND operation of both end points *A* and *B*.

6. If the result of AND operation is 0000, the line is partially visible and then find the point of intersection.

**4. a. Write at least four properties of Bezier curves. Calculate and also, roughly trace the Bezier curve for three control points (1,1), (2,2) and (3,1).**

**Ans.** **Properties of Bezier curves :**
1. Bezier curve always passes through the first and last control points *i.e.*, curve has same end points as the guiding polygon.
2. The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. Therefore, for four control points, the degree of the polynomial is three, *i.e.*, cubic polynomial.
3. The direction of the tangent vector at the end points is the same as that of the vector determined by first and last segments.
4. The curve lies entirely within the convex hull formed by four control points.
    **Numerical :**
    Control functions for point $(1, 1), (2, 2), (3, 1)$ are :
$$B_{0,2}(u) = {}^2C_0\, u^0\, (1-u)^{2-0}$$
$$= (1-u)^2$$
$$B_{1,2}(u) = {}^2C_1\, u^1(1-u)^{2-1}$$
$$= 2u(1-u)$$
$$B_{2,2}(u) = {}^2C_2\, u^2\, (1-u)^{2-2}$$
$$= u^2$$
    Bezier curve, $C(u) = p_0 B_{0,2}(u) + p_1 B_{1,2}(u) + p_2 B_{2,2}(u)$
    Dissolving curve in the $x, y$ coordinate :

$$x(u) = x_0\, B_{0,2}(u) + x_1 B_{1,2}(u) + x_2\, B_{2,2}(u)$$
$$y(u) = y_0 B_{0,2}(u) + y_1 B_{1,2}(u) + y_2 B_{2,2}(u)$$

where,

$x_0 = 1$      $y_1 = 1$
$x_1 = 2$      $y_2 = 2$
$x_2 = 3$      $y_3 = 1$

$$
\begin{aligned}
x(u) &= 1(1-u)^2 + 2[2u(1-u)] + 3u^2 \\
&= 1 + u^2 - 2u + 4u - 4u^2 + 3u^2 \\
&= 1 + 2u \\
y(u) &= 1\,(1-u)^2 + 2 \times [2u(1-u)] + 1u^2 \\
&= 1 + u^2 - 2u + 4u - 4u^2 + u^2 \\
&= 1 + 2u - 2u^2
\end{aligned}
$$



**Fig. 11.**

**b. Explain specular reflection and Phong model.**

**Ans.** **Specular reflection :**

1. Specular reflection is the result of total or near total reflection of the light in a concentrated region around the specular reflection angle.

2. When we see an illuminated shiny surface, like polished metal, shiny apple, or a pearl, then we observe a bright spot or a highlighted spot at a certain viewing direction.

3. In the Fig. 12(a) the specular reflection angle, denoted by $\theta$ is the same as the angle of incident light and $N$ is the unit normal surface vector, $R$ is the unit vector in the direction of ideal specular reflection, $L$ the unit vector directed towards the point light source, and $V$ is the unit vector pointing towards the viewer from the surface position.



**Fig. 12.** (a) Specular reflection angle = Angle of incidence.

**Fig. 12.** (b) Shiny (c) Dull surface.

4. Angle $\theta$ is the viewing angle relative to the specular reflection direction $R$.

5. In the case of an ideal reflector or perfect mirror, the incident light is reflected only in the specular reflection direction.

6. In this case, we would only see reflected light when vectors $V$ and $R$ coincide, *i.e.*, $\theta = 0$. The objects other than ideal reflectors have specular reflection around vector $R$.

   **Phong model :**

1. According to the Phong model, the intensity of specular reflection is proportional to $\cos^{n_s} \theta$. The value of $n$, specular reflection parameter, is determined by the type of surface.

2. For very shiny surfaces, the value of $n_s$ is large (more than 100) and for dull surfaces the value of $n_s$ is close to 1. For a perfect reflector, the value of $n_s$ is infinite.

3. The intensity of specular reflection depends on the angle of incidence, on the properties of the surface, colour of incident light and polarization.

4. The specular reflection coefficient $W(\theta)$, for each surface may be used to model the monochromatic specular intensity variation.

5. We can write the Phong specular reflection model as

$$I_{spec} = W(\theta)\, I_l \cos^{n_s} \theta \qquad \qquad ...(1)$$

   where $I_l$ is intensity of light source, and $\theta$ is the viewing angle relative to specular reflection direction $R$.

6. In the case of opaque materials, the specular reflection rejection is almost constant for all incidence angles. Therefore, replacing $W(\theta)$ with a constant specular reflection coefficient $K_s$. Since $V$ and $R$ are unit vectors in the viewing and specular reflection directions, $\cos \theta = V.R$. We can rewrite the equation (1) as

$$I_{spec} = K_s\, I_l\, (V.R)^{n_s}$$



**Fig. 13.** Projection of L on to the direction of normal vector N.

**c. Write back-face detection algorithm of hidden lines.**

**Ans.** **Back-face detection algorithm :**

1. Object surfaces that are orientated away from the viewer are called back-faces.

2. Algorithm used to detect the back-faces are known as back-face detection algorithm.

3. If any three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$ on any plane surface are known, the unknown parameters $A$, $B$, $C$ and $D$ of the plane surface equation $Ax + By + Cz + D = 0$ can be found as follows :

i. All the three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$ should satisfy the equation $Ax + By + Cz + D = 0$ as it lies on the surface.

Hence,
$$Ax_1 + By_1 + Cz_1 + D = 0$$
$$Ax_2 + By_2 + Cz_2 + D = 0$$
$$Ax_3 + By_3 + Cz_3 + D = 0$$

ii. Also, any arbitrary point $(x, y, z)$ lying on the surface should satisfy the equation of the desired surface

$$Ax + By + Cz + D = 0$$

iii. A unique solution of equations for $A$, $B$, $C$ and $D$ can only be obtained if

or $\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x & y & z & 1 \end{vmatrix} = 0$ (By Cramer's rule)

or $\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 & 0 \\ x_1 - x_2 & y_1 - y_2 & z_1 - z_2 & 0 \\ x_2 - x_3 & y_2 - y_3 & z_2 - z_3 & 0 \\ x & y & z & 1 \end{vmatrix} = 0$

or $\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ x_2 - x_3 & y_2 - y_3 & z_2 - z_3 \end{vmatrix} = 0$

or $(x - x_1) \begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} - (y - y_1) \begin{vmatrix} x_1 - x_2 & z_1 - z_2 \\ x_2 - x_3 & z_2 - z_3 \end{vmatrix}$

$$+ (z - z_1) \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} = 0$$

or $\begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} x + \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_1 - z_3 & x_2 - x_3 \end{vmatrix} y$

$$+ \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0$$

which is in the form of $Ax + By + Cz + D = 0$, giving the equation of the plane passing through the three points

where,

$$A = \begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix}$$

$$B = \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_2 - z_3 & x_2 - x_3 \end{vmatrix}$$

$$C = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix}$$

and

$$D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

iv. Any plane with its inside and outside surfaces can be defined by three points and its normal vector.

v. This normal vector can be calculated with three position vectors of three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$.

vi. Using right hand thumb rule, the three points are considered in counterclockwise manner for calculating the unit normal directing outside the surface. Thus, the surface normal and equation of the plane are calculated.

vii. Now, if any point $(x, y, \text{or } z)$ lies inside the polygon surface, $Ax + By + Cz + D < 0$, and if it is along the line of sight to the surface, the polygon must be back face.

viii. If $L$ is the line of sight or viewing vector (light vector in case of shadowing), perpendicular to the viewing plane originating from the eye or camera, $N$ is the unit surface normal vector as shown in Fig. 14($a$) and Fig. 14($b$) the polygon is the back face if $L.N > 0$



**Fig. 14.** (a) Geometry of camera vision.

ix. On aligning the light vector to that of the view vector which is normally along negative $z$-axis for right-handed viewing system, that is

$$L = 0i + 0j + L_z k$$
$$L.N = L_z K$$



**Fig. 14.** (b) Invisible and visible vision.

x. Hence, only $z$ component of normal vector $N$ is required to be considered, and the sign of $K$ is checked for negative value.

xi. Thus, any polygon is a back face if it has a surface normal with negative $z$ component value.

☺☺☺

# **B.Tech.**
# **(SEM. V) ODD SEMESTER THEORY**
# **EXAMINATION, 2013-14**
# **COMPUTER GRAPHICS**

**Time : 3 Hours** **Max. Marks : 100**

**Note :** Attempt **all** questions.

1. Attempt any **two** parts of the following : **(7 × 2 = 14)**

   a. **Explain frame buffer and video basics. Also, explain the terms pixel, aspect ratio, resolution.**

   b. **Develop and write mid-point circle algorithm. Apply it to predict the pixels in any octant for the circle whose centre is origin and radius = 14 units.**

   c. **Explain the parallel version of line algorithm by two methods.**

2. Attempt any **two** parts of the following : **(6 × 2 = 12)**

   a. **Reflect the polygon whose vertices are (–1, 0), (0, –2), (1, 0) and (0, 2) about the :**

   i. **Horizontal line $Y = 2$**

   ii. **Vertical line $X = 2$**

   b. **Write Cohen and Sutherland line clipping algorithm. Apply it for calculating the saved portion of a line from (2, 7) to (8, 12) in a window**
   **$(X_{w\min} = Y_{w\min} = 5$ and $X_{w\max} = Y_{w\max} = 10)$.**

   c. **Write Sutherland-Hodgeman polygon clipping algorithm. Explain the modification given by Weiler and Atherton for concave polygon.**

3. Attempt any **two** parts of the following : **(6 × 2 = 12)**

   a. **Derive rotation about *X*-axis, *Y*-axis and *Z*-axis matrices in 3D. Prove that for any rotation matrix**
   **$R^{-1}(\theta) = R(-\theta)$**

   b. **Derive oblique parallel projection and perspective projection matrices.**

   **c. Establish and write Cyrus-Beck 3D line clipping algorithm.**

**4.** Attempt any **two** parts of the following :                    **(6 × 2 = 12)**

   **a. Write at least four properties of Bezier curves. Calculate and roughly trace the Bezier curve for three control points (1, 1), (2, 2) and (3, 1).**

   **b. Explain specular reflection and Phong model.**

   **c. Write Depth-buffer method algorithm of hidden lines.**

☺☺☺

## SOLUTION OF PAPER (2013-14)

**Note :** Attempt **all** questions.

1. Attempt any **two** parts of the following :                    **(7 × 2 = 14)**
   a. **Explain frame buffer and video basics. Also, explain the terms pixel, aspect ratio, resolution.**

**Ans.** **Frame buffer :**
1. Frame buffer is a special area of memory in raster displays which is dedicated to graphics.
2. It holds the set of intensity values for all the screen points.
3. The stored intensity values are retrieved from frame buffer and displayed on the screen one row (scan line) at the time.
4. Each screen point is referred to as a pixel or pel (shortened forms of picture element).
5. Each pixel on the screen can be specified by its row and column number. Thus, by specifying row and column number we can specify the pixel position on the screen.

**Video basics :**
1. Video, or moving image in general, is created from a sequence of small images called frames.
2. By recording and then playing back frames in quick succession, an illusion of movement is created.
3. Video can be edited by removing some frames and combining sequences of frames, called clips, together in a timeline.

**Pixel :**
1. Pixel is the smallest part of the screen.
2. Each pixel has its own intensity, name or address by which we can control it.



Pixel

**Fig. 1.**

**Aspect ratio :**
1. An aspect ratio is an attribute that describes the relationship between the width and height of an image.
2. Aspect ratio is expressed by the symbolic notation *i.e.*, $x : y$.

**Resolution :**
1. Resolution is defined as the number of pixels on the horizontal axis and vertical axis.

2. The sharpness of the image on the display depends on the resolution and size of the monitor.

**b. Develop and write midpoint circle algorithm. Apply it to predict the pixels in any octant for the circle whose centre is origin and radius = 14 units.**

**Ans.** **Midpoint circle algorithm :**

1. Input radius $r$ and center $(x_c, y_c)$ and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$ *i.e.*, initialize starting position as

$$x = 0$$
$$y = r$$

2. Calculate the initial value of the decision parameter as

$$p = 1.25 - r$$

3. do

    {

        plot $(x, y)$

        if $(p < 0)$

          {

          $x = x + 1$

          $y = y$

          $p = p + 2x + 1$

          }

        else

          }

          $x = x + 1$

          $y = y - 1$

          $p = p + 2x - 2y + 1$

          }

        }

    while $(x < y)$

4. Determine symmetry points in the other seven octants.

5. Stop.

**Numerical :** $r = 14$

The initial point $(x, y) = (0, 14)$

*i.e.*, $x = 0$ and $y = 10$

Calculate initial decision parameters $p = 1 - r$

$$= 1 - 14 = -13$$
$$p = -13 (< 0)$$

First plot $(0, 14)$

Here $p < 0$ so $x = 0 + 1 + 1$

$$y = 14$$

and $p = -13 + 2 \times 1 + 1 = -13 + 3 = -10$

$x < y$ i.e., $1 < 14$ so, condition is true plot (1, 14)

Now $\qquad p = -10 (< 0)$

So $\qquad x = 1 + 1 = 2$

$\qquad\qquad y = 14$

and $\qquad p = -10 + 2 \times 2 + 1 = -10 + 5 = -5$

$x < y$ i.e., $2 < 14$ so, plot (2, 14)

Now $\qquad p = -5 (< 0)$

So $\qquad x = 2 + 1 = 3$

$\qquad\qquad y = 14$

$\qquad\qquad p = -5 + 2 \times 3 + 1 = -5 + 7 = 2$

$x < y$ so plot (3, 14)

Now $\qquad p = 2 (> 0)$

So $\qquad x = x + 1 = 3 + 1 = 4$

$\qquad\qquad y = y - 1 = 14 - 1 = 13$

$\qquad\qquad p = p + 2x - 2y + 1$

$\qquad\qquad\quad = 2 + 2 \times 4 - 2 \times 13 + 1$

$\qquad\qquad\quad = 2 + 8 - 26 + 1$

$\qquad\qquad\quad = 10 - 26 + 1 = -15$

$x < y$, so the next point to plot is (4, 13)

Now $\qquad p = -15 (< 0)$

$\qquad\qquad x = x + 1 = 4 + 1 = 5$

$\qquad\qquad y = 13$

$\qquad\qquad p = p + 2x + 1$

$\qquad\qquad\quad = -15 + 10 + 1 = -4$



**Fig. 2.**

$x < y$, so next point to plot is (5, 13)

Now $\qquad p = -4 (< 0)$

So $\qquad x = x + 1$

$\qquad\qquad\quad = 5 + 1 = 6$

$\qquad\qquad y = 13$

$\qquad\qquad p = p + 2x + 1$

$\qquad\qquad\quad = -4 + 12 + 1 = 8 + 1 = 9$

$x < y$, so plot point (6, 13)

Now $\qquad p = 9 (> 0)$

So $\qquad x = x + 1 = 6 + 1 = 7$

$\qquad\qquad y = y - 1 = 12$

$$p = 9 + 14 - 24 + 1$$
$$= 9 - 10 + 1 = 0$$

Note : we have to continue iteration till $x = y$ or $x > y$.

**c. Explain the parallel version of line algorithm by two methods.**

**Ans.** In the parallel version of line algorithm, pixel positions are calculated along a line path simultaneously by partitioning the computations among the various processors available. Methods for parallel version of line algorithm are :

**1. Bresenham's line algorithm :**

a. Given $n_p$ processors, a parallel Bresenham 's line algorithm can be used by subdividing the line path into $n_p$ partitions and simultaneously generating line segments in each of the sub-intervals.

b. For a line with slope $0 < m < 1$ and left endpoint coordinate $(x_0, y_0)$, the line is partitioned along the positive $x$-direction.

c. The distance between beginning $x$ positions of adjacent partitions is calculated as :

$$\Delta x_p = \frac{\Delta x + n_p - 1}{n_p}$$

where $\Delta x$ is the width of the line and the value for partition width $\Delta x_p$ is computed using integer division.

d. The starting $x$ coordinate is calculated for the $k$th partition as :
$$x_k = x_0 + k\Delta x_p$$

e. At the $k$th partition, the starting $y$ coordinate is :
$$y_k = y_0 + \text{round } (k\Delta y_p)$$

f. The initial decision parameter for Bresenham's algorithm at the start of the $k$th sub interval is :
$$p_k = (k\Delta x_p)\,(2\Delta y) - \text{round } (k\Delta y_p)\,(2\Delta x) + 2\Delta y - \Delta x$$

**2. In raster scan system :**

a. In raster systems we assign each processor to a particular group of screen pixel.

b. For lines with a positive slope greater than 1, we reverse the value of $x$ and $y$. *i.e.*, we sample at unit $y$ intervals ($\Delta y = 1$) and calculate each succeeding $x$ value as

$$x_{k+1} = x_k + \frac{1}{m}$$

c. Lines are to be processed from the left endpoint to the right endpoint. If this processing is reversed, so that the starting endpoint is at the right, then either we have $\Delta x = -1$ and

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

or (when the slope is greater than 1) we have $\Delta y = -1$ with
$$x_{k+1} = x_k +$$

d. Perpendicular distance $d$ from the line in to pixel as shown in Fig. 3 with coordinates $(x, y)$ is obtained with the calculation

$$d = Ax + By + C$$

where,

$$A = \frac{-\Delta y}{\text{linelength}}$$

$$B = \frac{\Delta x}{\text{linelength}}$$

$$C = \frac{x_0 \Delta y - y_0 \Delta x}{\text{linelength}}$$

$$\text{linelength} = \sqrt{\Delta x^2 + \Delta y^2}$$
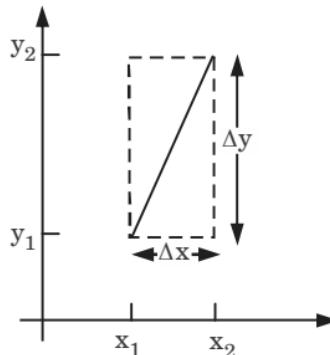


**Fig. 3.**

**2.** Attempt any **two** parts of the following : **(6 × 2 = 12)**

**a. Reflect the polygon whose vertices are (–1, 0), (0, –2), (1, 0) and (0, 2) about the :**

**i. Horizontal line $Y = 2$**

**ii. Vertical line $X = 2$**

**Ans.** **Polygon is represented by homogeneous coordinates matrix as :**

$$V = [ABCD] \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}$$

i. $$Y = 2$$

On comparing with $y = mx + c$

$$m = 0, c = 2.$$

$$\theta = \tan^{-1}(m) = \tan^{-1}(0) = 0$$

The operation of reflection through an arbitrary line (not passing through origin) is required to follow the following procedures :

$$[T] = [T_{\text{trans}}]\ [R_\theta][R_{\text{ref}}][R_\theta]^{-1}[T_{\text{trans}}]^{-1}$$

$$[T_{\text{trans}}] = \text{Translation matrix}$$

$[R_\theta]$ = Rotation matrix

$[R_{ref}]$ = Reflection matrix

$[R_\theta]^{-1}$ = Inverse rotation matrix

$[T_{trans}]^{-1}$ = Inverse translation matrix

$$[T_{trans}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -c & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 0 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos(\theta) & \sin(-\theta) & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{ref} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[R_\theta]^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_{trans}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & c & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\text{Reflected polygon} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 2 & 0 \\ 0 & 4 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Now coordinates of polygon is, $(-1, 2)$, $(0, 4)$, $(1, 2)$ and $(0, 0)$.

ii. $\qquad\qquad X = 2$

$$y = mx + c$$

$$x = \frac{y}{m} - \frac{c}{m}$$

On comparing $m = \infty$, $c = -2$

$$\theta = \tan^{-1}(m) = \tan^{-1}(\infty) = 90°$$

$$[T_{\text{trans}}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos(-90°) & \sin(-90°) & 0 \\ -\sin(-90°) & \cos(-90) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{\text{ref}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_{\text{trans}}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

$$[R_\theta]^{-1} = \begin{bmatrix} \cos 90° & \sin 90° & 0 \\ -\sin(-90°) & \cos(-90°) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T] = [T_{\text{trans}}][R_\theta][R_{\text{ref}}][R_\theta]^{-1}[T_{\text{trans}}]^{-1}$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\text{Reflected polygon} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & 2 & 0 \end{bmatrix}$$

Hence, new coordinates of polygon is,
(0, 1), (– 2, 2), (0, 3) and (2, 2).

**b. Write Cohen and Sutherland line clipping algorithm. Apply it for calculating the saved portion of a line from (2, 7) to (8, 12) in a window**
$$(X_{w\min} = Y_{w\min} = 5 \text{ and } X_{w\max} = Y_{w\max} = 10).$$

**Ans.** **Cohen and Sutherland algorithm :**

1. In this algorithm, a rectangular window is considered whose coordinates are $(xw_{\min}, yw_{\min}, xw_{\max}, yw_{\max})$ as shown in Fig. 4.

**Fig. 4.**

2. Every line end point is assigned a 4-digit binary code (region code) that identifies the location of the point corresponding to the boundaries of the clipping window.

3. The procedure for calculating the end point $p(x, y)$ code is as follow :

**a. For bit 1**

If $(y - yw_{max}) < 0$ then
code = 0;
else
code = 1;

**b. For bit 2**

If $(yw_{min} - y) < 0$ then
code = 0;
else
code = 1;

**c. For bit 3**

if$(x - xw_{max}) < 0$ then
code = 0;
else
code = 1;

**d. For bit 4**

if$(xw_{min} - x) < 0$ then
code = 0;
else
code = 1;

4. If the region code of both the end points of a line segment is 0000, the line is completely visible as it completely lies within the window.

5. If the region code of both the end points of the line segment is not 0000 then we calculate AND operation of both end points *A* and *B*.

6. If the result of AND operation is 0000, the line is partially visible and then find the point of intersection.

**Numerical :**

Let $P_1 = (2, 7)$ and $P_2 = (8, 12)$, windows lower left corner = $(5, 5)$ windows upper right corner = $(10, 10)$ as shown in Fig. 5.

**Fig. 5.**

Now, we assign the four digit binary code for $P_1$ and $P_2$ shown in Table 1.

**Table 1.**

|  |  | For $P_1$ (2, 7) | For $P_2$ (8, 12) |
|---|---|---|---|
| Bit 1 : $\quad x - x_{w\min}$ | $x - 5$ | $-3$ | 3 |
| Bit 2 : $\quad x_{w\max} - x$ | $10 - x$ | 8 | 2 |
| Bit 3 : $\quad y - y_{w\min}$ | $Y - 5$ | 2 | 7 |
| Bit 4 : $\quad y_{w\max} - y$ | $10 - y$ | 3 | $-2$ |

If sign bit is negative, it will assign unity(1) and if it is positive, it will assign zero(0).

Binary code for $P_1(2, 7)$ : | bit 4 | bit 3 | bit 2 | bit 1 | 0001

Binary code for $P_2(8, 12)$ : | bit 4 | bit 3 | bit 2 | bit 1 | 1000

So, AND of $P_1$ and $P_2$ = 0000

Line $P_1P_2$ intersect the window at $Q(x_1, y_1)$ and $R(x_2, y_2)$. To find the intersection points we use equation of line $P_1P_2$.

Slope of line $P_1P_2$ is $m = \dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{12 - 7}{8 - 2} = \dfrac{5}{4}$

Equation of line $P_1P_2$ is given as

$$y = mx + c$$

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right) x + c$$

$$y = \frac{5}{4}x + c \qquad \qquad ...(1)$$

$\because$ $P(2, 7)$ lies on the line so it satisfies the equation (1)

Put $x = 2$ and $y = 7$ in equation (1)

$$7 = \frac{5}{4} \times 2 + c$$

$$c = 7 - \frac{5}{2} = \frac{9}{2} = 4.5$$

Equation of line $P_1P_2$ is given as

$$y = \frac{5}{4} x + 4.5 \qquad \qquad ...(2)$$

**For intersection point $Q(x_1, y_1)$ :**

Since, $Q(x_1, y_1)$ lies on line and the boundary of window. So, it satisfy the equations (2) and $x_1 = 5$

$$y_1 = \frac{5}{4} x_1 + 4.5$$

$$4y_1 - 5x_1 = 18$$

Put $x_1 = 5$, we get

$$4y_1 = 18 + 5 \times 5$$

$$y_1 = \frac{18 + 25}{4} = \frac{43}{4} = 10.75$$

**For intersection point $R(x_2, y_2)$ :**

Since, $R(x_2, y_2)$ lies on line and the boundary of window. So, $y_2 = 10$

$$y_2 = \frac{5}{4} x_2 + 4.5$$

$$4y_2 - 5x_2 = 18$$

Put $x_2 = 10$

$$40 - 5x_2 = 18$$

$$x_2 = \frac{40 - 18}{5} = 4.4$$

So, the point of intersection are (5, 10.75) and $R$(4.4, 10)

Visible line segment is $Q$(5, 10.75) to $R$(4.4, 10)

**c. Write Sutherland-Hodgeman polygon clipping algorithm. Explain the modification given by Weiler and Atherton for concave polygon.**

**Ans.** **Sutherland-Hodgeman polygon clipping algorithm :**

1. Read coordinates of all vertices of the polygon.
2. Read coordinates of the clipping window.
3. Consider the left edge of the window.
4. Compare the vertices of each edge of the polygon, individually with the clipping plane.
5. Save the resulting intersections and vertices in the new list of vertices according to four possible cases between the edge and the clipping boundary.
6. Repeat the step 4 and 5 for remaining edges of the clipping window. Each time the resultant list of vertices is successively passed, move the process to the next edge of the clipping window.
7. Stop.

### Weiler-Atherton polygon clipping :

1. In this algorithm, the vertex processing procedure is modified so that the concave polygons are displayed correctly.
2. This algorithm depends on identifying surfaces as shown in Fig. 6.



Clipping a concave polygon by          Output (two separate
applying Weiler-Atherton algorithm          polygon areas)

**Fig. 6.**

3. There are two directions (clockwise or anticlockwise) that exist to process the polygon vertices.
4. For clockwise processing of polygon vertices, we use the following rules :
   a. For an outside to inside pair of vertices, follow the polygon boundary.
   b. For an inside to outside pair of vertices, follow the window boundary in a clockwise direction.

**3.** Attempt any **two** parts of the following :                    **(6 × 2 = 12)**

**a. Derive rotation about *X*-axis, *Y*-axis and *Z*-axis matrices in 3D. Prove that for any rotation matrix**
$$R^{-1}(\theta) = R\ (-\ \theta)$$

**Ans.**

**i. Rotation about *x*-axis (*i.e.* axis of rotation is *x*) :**



**Fig. 7.** Rotation when axis of rotation is *x*-axis.

$$x' = x$$
$$y' = r \cos(\alpha + \theta)$$
$$y' = r \cos\alpha \cos\theta - r \sin\alpha \sin\theta \qquad ...(1)$$

Since $\qquad r \cos\alpha = y$ and $r \sin\alpha = z$

Put the value in eq. (1) we get,

$$y' = y \cos\theta - z \sin\theta$$
$$z' = r \sin(\alpha + \theta)$$

or, $\qquad\qquad = r \sin\alpha \cos\theta + r \cos\alpha \sin\theta$

$\qquad\qquad\qquad = z \cos\theta + y \sin\theta$

or, $\qquad\qquad R_{\theta,i} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$
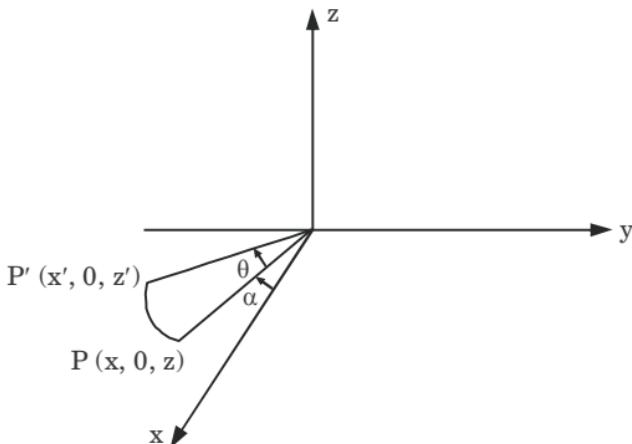
The points after rotation can be calculated by the following equation :

$$p'(x', y', z') = R_{\theta,i} P(x, y, z)$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**ii. Rotation about *y*-axis (*i.e.*, *y*-axis is axis of rotation) :**



**Fig. 8.**

$$x' = x \cos\theta + z \sin\theta$$
$$y' = y$$
$$z' = -x \sin\theta + z \cos\theta$$
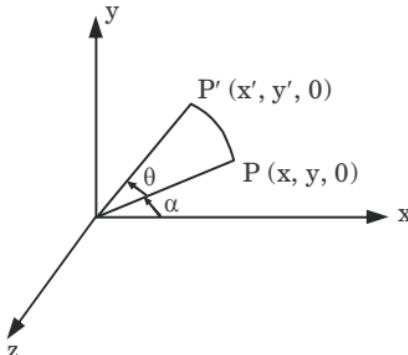
or, $\qquad\qquad R_{\theta,i} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**iii. Rotation about the $z$-axis (*i.e.*, $z$-axis is axis of rotation)**



**Fig. 9.**

$x' = x\cos\theta - y\sin\theta$
$y' = z\cos\theta + y\sin\theta$
$z' = z$

or, $\qquad R_{\theta,k} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Numerical :**

Let $\qquad R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Put $\theta = (-\theta)$, we get

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) & 0 \\ 0 & \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since $\cos(-\theta) = \cos(\theta)$, $\sin(-\theta) = -\sin\theta$, we have

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now,

$$R(-\theta) * R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

We have,

$$R(-\theta) . R(\theta) = I$$

Multiply $R^{-1}(\theta)$ both sides,

$$R(-\theta).R(\theta).R^{-1}(\theta) = I.R^{-1}(\theta)$$
$$R(-\theta) = R^{-1}(\theta)$$

Hence proved.

**b. Derive oblique parallel projection and perspective projection matrices.**

**Ans.** **Derivation of oblique parallel projection :**

1. An oblique projection is obtained by projecting points along parallel lines that are not perpendicular to the projection plane.
2. An oblique projection of a point $P(x, y, z)$ on the $z = 0$ plane is shown in Fig. 10.
3. $P_{ob} (x_p, y_p)$ is oblique projection of the point $P$ and $P_{or} (x, y)$ is the orthographic projection of $P$ on the $z = 0$ plane.



**Fig. 10.**

4. Let $L$ be the length of the line joining points $P_{ob}$ $(x_p, y_p)$ and $P_{or}$ $(x, y)$, $\theta$ be the angle made by the line $L$ with $x$-axis.

5. We can express the projection coordinates in terms of $x, y, L$ and $\theta$ as

$$x_p = x + L \cos \theta$$
$$y_p = y + L \sin \theta$$

6. Length $L$ depends on the angle $\alpha$ and the $Z$ coordinate of the point to be projected

$$\tan \alpha = \frac{Z}{L} \Rightarrow L = \frac{Z}{\tan \alpha} = ZL_1 \quad \left( \text{where } L_1 = \frac{1}{\tan \alpha} \right)$$

when $\quad Z = 1, L = L_1$

7. Now, we can write the oblique projection as
$$x_p = x + Z(L_1 \cos \theta)$$
$$y_p = x + Z(L_1 \sin \theta)$$

8. Considering $\quad Z_p = 0$, we get

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L_1 \cos \theta & 0 \\ 0 & 1 & L_1 \sin \theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Derivation of perspective projection matrix :**

1. Let us consider the center of projection is at $(x_c, y_c, z_c)$ and the point on object is $(x_1, y_1, z_1)$, then the parametric equation for the line containing these points can be given as

$$x_2 = x_c + (x_1 - x_c)u$$
$$y_2 = y_c + (y_1 - y_c)u$$
$$z_2 = z_c + (z_1 - z_c)u$$

where $u$ is the parameter

2. For projected point $z_2$ is 0, therefore, the third equation can be written as
$$0 = z_c + (z_1 - z_c)u$$

$$\therefore \qquad u = - \frac{z_c}{(z_1 - z_c)}$$

3. Put this value of $u$ in first two equations, we get

$$x_2 = x_c - \frac{z_c}{(z_1 - z_c)} (x_1 - x_c)$$

$$= \frac{x_c z_1 - x_c z_c - x_1 z_c + x_c z_c}{(z_1 - z_c)}$$

$$\therefore \qquad x_2 = \frac{x_c z_1 - x_1 z_c}{(z_1 - z_c)}$$

and
$$y_2 = y_c - \frac{z_c}{(z_1 - z_c)} (y_1 - y_c)$$

$$= \frac{y_c z_1 - y_c z_c - y_1 z_c + y_c z_c}{(z_1 - z_c)}$$

$$y_2 = \frac{y_c z_1 - y_1 z_c}{(z_1 - z_c)}$$

4. So, the homogeneous matrix can be written as

$$[x_2 \quad y_2 \quad z_2 \quad 1] = [x_1 \quad y_1 \quad z_1 \quad 1] \begin{bmatrix} -z_c & 0 & 0 & 0 \\ 0 & -z_c & 0 & 0 \\ x_c & y_c & 0 & 1 \\ 0 & 0 & 0 & -z_c \end{bmatrix}$$

**c. Establish and write Cyrus-Beck 3D line clipping algorithm.**

**Ans.**

1. Cyrus-Beck algorithm is applicable to an arbitrary convex region.

2. This algorithm uses a parametric equation of line segment to find the intersection points of a line with the clipping edges.

3. The parametric equation of a line segment from $P_1$ to $P_2$ is

$$P(t) = P_1 + (P_2 - P_1) t \qquad ; 0 \le t \le 1$$

where $t$ is a parameter, $t = 0$ at $P_1$ and $t = 1$ at $P_2$.

4. Consider a convex clipping region $R$, $f$ is a boundary point of the convex region $R$ and $n$ is an inner normal for one of its boundaries, as shown in Fig. 11.



**Fig. 11.** Convex region, boundary point and inner normal.

5. We evaluate the dot product $n \cdot [P(t) - f]$, and check the following condition as :

a. If dot product is negative, $i.e.$,
$$n \cdot [P(t) - f] < 0$$
Then the vector $P(t) - f$ is pointed away from the interior of $R$.

    b.    If dot product is zero, *i.e.*,
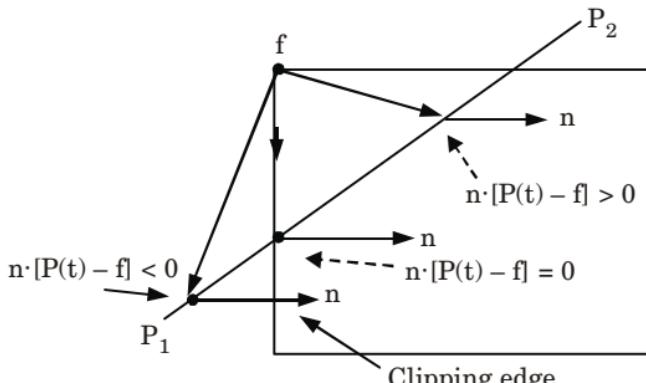
$$n \cdot [P(t) - f] = 0$$

        Then $P(t) - f$ is pointed parallel to the plane containing $f$ and perpendicular to the normal.

    c.    If dot product is positive, *i.e.*,

$$n \cdot [P(t) - f] > 0$$

        Then the vector $P(t) - f$ is pointed towards the interior of $R$, as shown in Fig. 12.



**Fig. 12.** Dot product for three inside, outside and on the boundary of the clipping region.

**Cyrus-Beck line clipping algorithm :**

1. Read two endpoints of the line, $P_1$ and $P_2$.

2. Read vertex coordinates of the clipping window.

3. Calculate $D = P_2 - P_1$.

4. Assign boundary point ($f$) with particular edge.

5. Find inner normal vector for corresponding edge.

6. Calculate $D \cdot n$ and $W = P_1 - f$.

7. If $D \cdot n > 0$.

$$t_L = -\frac{W \cdot n}{D.n}$$

    else

$$t_U = -\frac{W \cdot n}{D.n}$$

    end if

8. Repeat steps 4 to 7 for each edge of the clipping window.

9. Find maximum lower limit and minimum upper limit.

10. If maximum lower limit and minimum upper limit do not satisfy condition $0 \leq t \leq 1$ then ignore the line.

11. Calculate the intersection points by substituting values of maximum lower limit and minimum upper limit in the parametric equation of the line $P_1 P_2$.

12. Draw the line segment $P(t_L)$ to $P(t_U)$.

13. Stop.

**4.** Attempt any **two** parts of the following : **(6 × 2 = 12)**

**a. Write at least four properties of Bezier curves. Calculate and roughly trace the Bezier curve for three control points (1, 1), (2, 2) and (3, 1).**

**Ans. Properties of Bezier curves :**

1. Bezier curve always passes through the first and last control points *i.e.*, curve has same end points as the guiding polygon.

2. The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. Therefore, for four control points, the degree of the polynomial is three, *i.e.*, cubic polynomial.

3. The direction of the tangent vector at the end points is the same as that of the vector determined by first and last segments.

4. The curve lies entirely within the convex hull formed by four control points.

**Numerical :**

Control functions for point $(1, 1), (2, 2), (3, 1)$ are :

$$B_{0,2}(u) = {}^2C_0 \, u^0 \, (1-u)^{2-0}$$
$$= (1-u)^2$$
$$B_{1,2}(u) = {}^2C_1 \, u^1(1-u)^{2-1}$$
$$= 2u(1-u)$$
$$B_{2,2}(u) = {}^2C_2 \, u^2 \, (1-u)^{2-2}$$
$$= u^2$$

Bezier curve, $C(u) = p_0 B_{0,2}(u) + p_1 B_{1,2}(u) + p_2 B_{2,2}(u)$

Dissolving curve in the $x, y$ coordinate :

$$x(u) = x_0 B_{0,2}(u) + x_1 B_{1,2}(u) + x_2 B_{2,2}(u)$$
$$y(u) = y_0 B_{0,2}(u) + y_1 B_{1,2}(u) + y_2 B_{2,2}(u)$$

where, 
$$x_0 = 1 \qquad\qquad y_1 = 1$$
$$x_1 = 2 \qquad\qquad y_2 = 2$$
$$x_2 = 3 \qquad\qquad y_3 = 1$$
$$x(u) = 1(1-u)^2 + 2[2u(1-u)] + 3u^2$$
$$= 1 + u^2 - 2u + 4u - 4u^2 + 3u^2$$
$$= 1 + 2u$$
$$y(u) = 1 (1-u)^2 + 2 \times [2u(1-u)] + 1u^2$$
$$= 1 + u^2 - 2u + 4u - 4u^2 + u^2$$
$$= 1 + 2u - 2u^2$$

**Fig. 13.**

b. **Explain specular reflection and Phong model.**

**Ans.** **Specular reflection :**

1. Specular reflection is the result of total or near total reflection of the light in a concentrated region around the specular reflection angle.

2. When we see an illuminated shiny surface, like polished metal, shiny apple, or a pearl, then we observe a bright spot or a highlighted spot at a certain viewing direction.

3. In the Fig. 14($a$) the specular reflection angle, denoted by $\theta$ is the same as the angle of incident light and $N$ is the unit normal surface vector, $R$ is the unit vector in the direction of ideal specular reflection, $L$ the unit vector directed towards the point light source, and $V$ is the unit vector pointing towards the viewer from the surface position.

4. Angle $\theta$ is the viewing angle relative to the specular reflection direction $R$.

5. In the case of an ideal reflector or perfect mirror, the incident light is reflected only in the specular reflection direction.



**Fig. 14.** (a) Specular reflection angle = Angle of incidence.



**Fig. 14.** (b) Shiny (c) Dull surface.

6. In this case, we would only see reflected light when vectors *V* and *R* coincide, *i.e.*, θ = 0. The objects other than ideal reflectors have specular reflection around vector *R*.

   **Phong model :**

1. According to the Phong model, the intensity of specular reflection is proportional to $\cos^{n_s} \theta$. The value of *n*, specular reflection parameter, is determined by the type of surface.

2. For very shiny surfaces, the value of $n_s$ is large (more than 100) and for dull surfaces the value of $n_s$ is close to 1. For a perfect reflector, the value of $n_s$ is infinite.

3. The intensity of specular reflection depends on the angle of incidence, on the properties of the surface, colour of incident light and polarization.

4. The specular reflection coefficient W(θ), for each surface may be used to model the monochromatic specular intensity variation.

5. We can write the Phong specular reflection model as

$$I_{spec} = W(\theta)\, I_l \cos^{n_s} \theta \qquad ...(1)$$

   where $I_l$ is intensity of light source, and θ is the viewing angle relative to specular reflection direction *R*.

6. In the case of opaque materials, the specular reflection rejection is almost constant for all incidence angles. Therefore, replacing W(θ) with a constant specular reflection coefficient $K_s$. Since *V* and *R* are unit vectors in the viewing and specular reflection directions, $\cos \theta = V.R$. We can rewrite the equation (1) as

$$I_{spec} = K_s I_l (V.R)^{n_s}$$



**Fig. 15.** Projection of L on to the direction of normal vector N.

**c. Write Depth-buffer method algorithm of hidden lines.**

**Ans.** **Depth-buffer method or *z*-buffer algorithm :**

1. Depth-buffer method is a simple extension of the frame buffer idea. A frame is used to store the intensity of each pixel in image space.

2. The depth-buffer method is used to store the *z*-coordinates or depth or every visible pixel in image space.

3. In this algorithm a buffer of the same size as the frame buffer, is set up which holds depth information.

4. Each elements of the depth buffer corresponds to a pixel in the frame buffer and initially holds the maximum depth in the scene.

5. As each polygon is scan-converted, the depth at each pixel is calculated and compared with the corresponding depth in the depth buffer.

6. If the depth is less than that stored in the depth buffer (*i.e.*, nearer the viewer) then that pixel is set in the frame buffer with the polygon colour at that point and the depth buffer is set to the polygon depth.

7. If the polygon depth is greater (*i.e.*, further away from the viewer) than the depth buffer at that point is not stored in the frame buffer.

8. The depth-buffer algorithm can be stated as :

   **Step 1 :** Initialize frame buffer to background colour.

   **Step 2 :** Initialize depth-buffer to minimum $z$ value.

   **Step 3 :** Scan-convert each polygon in arbitrary order.

   **Step 4 :** For each $(x, y)$ pixel, calculated depth '$z$' at that pixel $(z(x, y))$.

   **Step 5 :** Compare calculated new depth $z(x, y)$ with value previously stored in depth-buffer at the location $z(x, y)$.

   **Step 6 :** If $z(x, y) > z(x, y)$, then write the new depth value to depth-buffer and updated frame buffer.

   **Step 7 :** Otherwise, no action is taken.

☺☺☺

# B.Tech.
## (SEM. IV) EVEN SEMESTER THEORY EXAMINATION, 2014-15
## COMPUTER GRAPHICS

**Time : 2 Hours**        **Max. Marks : 50**

**Note :** Attempt **all** questions.

1. Attempt any **two** parts of the following :     **(6 × 2 = 12)**
   a. **Write merits and demerits of LCD (Liquid Crystal Display).**

   b. **If base address of a frame buffer is 100 and screen size is (15 inch × 19 inch) with resolution 13 dpi (dot per inch) calculate the memory location where the coordinate of pixels are stored.**
   i. **Pixel $P1$ at $A$ (200, 25)**
   ii. **Pixel $P2$ at $B$(75, 45)**

   c. **Calculate value of pixels which is scan converted by DDA algorithm and endpoint of line is (4, 4) and (– 3, 0).**

2. Attempt any **two** parts of the following :     **(6 × 2 = 12)**
   a. **Write Bresenham's algorithm of line and explain.**

   b. **Explain the Sutherland-Cohen line clipping algorithm.**

   c. **Differentiate between raster and vector graphics.**

3. Answer any **two** parts of the following :     **(6 × 2 = 12)**
   a. **Explain Phong and Warn model for illumination.**

   b. **Let $P_0$ (0, 0), $P_1$ (1, 2), $P_2$ (2, 1), $P_3$ (3, 1), $P_4$ (4, 10) and $P_5$ (5, 5) be given data control points. If interpolation based on Bezier curve is used to find a curve interpolating these data points. Find parametric mid-point of the gradient and also calculate coordinate of parametric quartiles of the curve.**

   c. **Explain various types of quadratic surfaces with their equation and blobby objects.**

**4.** Answer any **two** parts of the following :                    **(7 × 2 =14)**

**a.** **What do you understand by the term "Concatenation of transformations" ? What are its advantages ? If *A* and *B* are two different transformations, illustrate with suitable example that $A \cdot B \neq B \cdot A$.**

**b.** **Apply the Liang-Barsky to clip the line segment from *A* (3,7) to *B* (8,10) against the regular rectangular window *P* (1,2), *Q* (9,2), *R* (9,8) and *S* (1, 8).**

**c.** **Explain mid-point circle algorithm.**

☺☺☺

# SOLUTION OF PAPER (2014-15)

**Note :** Attempt **all** questions.

**1.** Attempt any **two** parts of the following : **(6 × 2 = 12)**

**a. Write merits and demerits of LCD (Liquid Crystal Display).**

**Ans. Merits of LCD :**
  i. It produces very bright images due to high peak intensity.
  ii. It produce lower electric, magnetic and electromagnetic fields than CRTs.
  iii. It has no geometric distortion at the native resolution. Minor distortion can occur for other resolutions.
  iv. It consumes less electricity than a CRT and produce little heat.

  **Demerits of LCD :**
  i. The aspect ratio and resolution are fixed.
  ii. It is not good at producing black and very dark grays levels.
  iii. It has lower contrast than CRTs due to a poor black-level.
  iv. Images are satisfactory, but not accurate as colour saturation is reduced at low intensity levels due to a poor black-level.

**b. If base address of a frame buffer is 100 and screen size is (15 inch × 19 inch) with resolution 13 dpi (dot per inch) calculate the memory location where the coordinate of pixels are stored.**

  **i. Pixel $P1$ at $A$ (200, 25)**

  **ii. Pixel $P2$ at $B$ (75, 45)**

**Ans.** Base address = 100

Resolution of screen = 195 * 247 pixels (or dots)

  i. Memory location of $P1$ = Base address + (195*200 + 25) = 39125
  ii. Memory location of $P2$ = Base address + (195*75 + 45) = 14770

**c. Calculate value of pixels which is scan converted by DDA algorithm and endpoint of line is (4, 4) and (– 3, 0).**

**Ans.**
$$(x_1, y_1) = (-3, 0) \quad , \quad (x_2, y_2) = (4, 4)$$
$$\Delta x = 4 - (-3) = 7$$
$$\Delta y = 4 - 0 = 4$$
$$\text{Length} = \Delta x \qquad\qquad (\because \Delta x > \Delta y)$$

$$\Delta x = \frac{4 - (-3)}{7} = 1$$

$$\Delta y = \frac{4 - 0}{7} = \frac{4}{7} = 0.57$$

$$x = x_1 + 0.5 \times \text{sign}(\Delta x) = -3 + 0.5 \times 1 = -2.5$$

$$y = y_1 + 0.5 \times \text{sign}(\Delta y)$$

$$= 0 + 0.5 \times 0.57 = 0.28$$

Plot integer $(-2.5)$ and $(0.28)$, *i.e.*, $(-3, 0)$

$$x = x + \Delta x = -2.5 + 1 = -1.5$$
$$y = 0.28 + 0.57 = 0.85$$

Plot integer $(-1.5)$ and $(0.85)$, *i.e.*, $(-2, 1)$

$$x = -1.5 + 1 = -0.5$$
$$y = 0.85 + 0.57 = 1.42$$

Plot integer $(-0.5)$ and $(1.42)$, *i.e.*, $(-1, 1)$

$$x = -0.5 + 1 = 0.5$$
$$y = 1.42 + 0.57 = 1.99$$

Plot integer $(0.5)$ and $(1.99)$, *i.e.*, $(1, 2)$

$$x = 0.5 + 1 = 1.5$$
$$y = 1.99 + 0.57 = 2.56$$

Plot integer $(1.5)$ and $(2.56)$, *i.e.*, $(2, 3)$

$$x = 1 + 1.5 = 2.5$$
$$y = 2.56 + 0.57 = 3.13$$
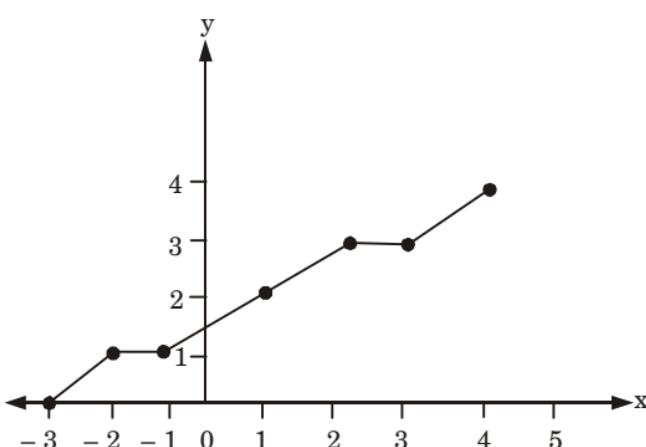
Plot integer $(2.5)$ and $(3.13)$, *i.e.*, $(3, 3)$

$$x = 2.5 + 1 = 3.5$$
$$y = 3.13 + 0.57 = 3.7$$

Plot integer $(3.5)$ and $(3.7)$, *i.e.*, $(4, 4)$



**Fig. 1.**

2. Attempt any **two** parts of the following :                    **(6 × 2 = 12)**
a. **Write Bresenham's algorithm of line and explain.**

**Ans.** **Bresenham's line drawing algorithm :**
1. Input the two lines endpoints and store the left endpoint $(x_0, y_0)$.
2. Load $(x_0, y_0)$ into frame buffer, *i.e.*, plot the first point.
3. Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for decision parameter as :

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

4. At each $x_k$ along the line, starting at $k = 0$, perform the following test :
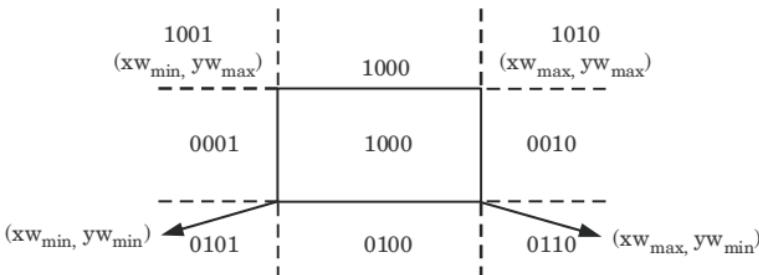
    a.   If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and
$$p_{k+1} = p_k + 2\Delta y$$
    b.   If $p_k > 0$, the next point to plot is $(x_k + 1, y_k + 1)$ and
$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$
5. Repeat step 4  $\Delta x$ times.

**b. Explain the Sutherland-Cohen line clipping algorithm.**

**Ans.** **Cohen and Sutherland algorithm :**

1. In this algorithm, a rectangular window is considered whose coordinates are $(xw_{min}, yw_{min}, xw_{max}, yw_{max})$ as shown in Fig. 2.



**Fig. 2.**

2. Every line end point is assigned a 4-digit binary code (region code) that identifies the location of the point corresponding to the boundaries of the clipping window.

3. The procedure for calculating the end point $p(x, y)$ code is as follow :

**a. For bit 1**
If $(y - yw_{max}) < 0$ then
code = 0;
else
code = 1;

**b. For bit 2**
If $(yw_{min} - y) < 0$ then
code = 0;
else
code = 1;

**c. For bit 3**
if$(x - xw_{max}) < 0$ then
code = 0;
else
code = 1;

**d. For bit 4**
if$(xw_{min} - x) < 0$ then
code = 0;
else
code = 1;

4. If the region code of both the end points of a line segment is 0000, the line is completely visible as it completely lies within the window.

5. If the region code of both the end points of the line segment is not 0000 then we calculate AND operation of both end points *A* and *B*.

6. If the result of AND operation is 0000, the line is partially visible and then find the point of intersection.

**c. Differentiate between raster and vector graphics.**

**Ans.**

| S. No. | Raster graphics | Vector graphics |
|--------|-----------------|-----------------|
| 1. | In raster graphics, the beam is moved all over the screen. | In random (vector) graphics, the beam is moved between the end points of the graphics primitive. |
| 2. | Scan conversion is required. | Scan conversion is not required. |
| 3. | It display smooth lines by approximating them with pixels on the raster grid. | It draws a continous and smooth line. |
| 4. | Cost is low. | Cost is more. |
| 5. | Raster graphics has ability to display areas filled with solid colours or patterns. | It only draws lines and characters. |

**3.** Answer any **two** parts of the following :                   **(6 × 2 = 12)**

**a. Explain Phong and Warn model for illumination.**

**Ans.** **Phong model :**

1. According to the Phong model, the intensity of specular reflection is proportional to $\cos^{ns} \theta$. The value of $n$, specular reflection parameter, is determined by the type of surface.

2. For very shiny surfaces, the value of $n_s$ is large (more than 100) and for dull surfaces the value of $n_s$ is close to 1. For a perfect reflector, the value of $n_s$ is infinite.

3. The intensity of specular reflection depends on the angle of incidence, on the properties of the surface, colour of incident light and polarization.

4. The specular reflection coefficient $W(\theta)$, for each surface may be used to model the monochromatic specular intensity variation.

5. We can write the Phong specular reflection model as

$$I_{spec} = W(\theta) \, I_l \cos^{ns} \theta \qquad \ldots(1)$$

where $I_l$ is intensity of light source, and $\theta$ is the viewing angle relative to specular reflection direction $R$.

6. In the case of opaque materials, the specular reflection rejection is almost constant for all incidence angles. Therefore, replacing $W(\theta)$ with a constant specular reflection coefficient $K_s$. Since $V$ and $R$ are

unit vectors in the viewing and specular reflection directions, $\cos\theta = V.R$. We can rewrite the equation (1) as

$I_{\text{spec}} = K_s I_l (V.R)^{ns}$



**Fig. 3.** Projection of L on to the direction of normal vector N.

**Warn model :**

1. The Warn model provides a method for simulating studio lighting effects by controlling light intensity in different directions.
2. Light sources are modeled as points on a reflecting surface, using the Phong model for the surface points. Then, the intensity in different directions is controlled by selecting values for the Phong exponent.
3. In addition, light controls, such as "barn doors" and spotlighting, used by studio photographers can be simulated in the Warn model.
4. Flaps are used to control the amount of light emitted by a source in various directions.
5. Two flaps are provided for each of the $x$, $y$ and $z$-directions.
6. Spotlights are used to control the amount of light emitted within a cone with apex at a point-source position.

**b.** **Let $P_0$ (0, 0), $P_1$ (1, 2), $P_2$ (2, 1), $P_3$ (3, 1), $P_4$ (4, 10) and $P_5$ (5, 5) be given data control points. If interpolation based on Bezier curve is used to find a curve interpolating these data points. Find parametric midpoint of the gradient and also calculate coordinate of parametric quartiles of the curve.**

**Ans.** Given points : $P_0$ (0, 0), $P_1$ (1, 2), $P_2$ (2, 1), $P_3$ (3, 1), $P_4$ (4, 10) and $P_5$ (5, 5)

$$u = 0.5$$
$$n + 1 = 6$$
$$n = 5$$



**Fig. 4.**

$$P(u) = \sum_{k=0}^{5} P_k \, BEZ_{k,n}(u)$$

$$BEZ_{k,5}(u) = C(5, k) \, u^k \, (1-u)^{5-k}$$

$$C_{5,k} = \frac{\lfloor 5}{\lfloor k \, \lfloor 5-k}$$

$$B_{0,5}(u) = C_{5,0} \, u^0 (1-u)^{5-0} = (1-u)^5$$
$$B_{1,5}(u) = C_{5,1} \, u^1 (1-u)^4 = 5(u) \, (1-u)^4$$
$$B_{2,5}(u) = C_{5,2} \, u^2 (1-u)^3 = 10u^2 \, (1-u)^3$$
$$B_{3,5}(u) = C_{5,3} \, u^3 (1-u)^2 = 10u^3 \, (1-u)^2$$
$$B_{4,5}(u) = C_{5,4} \, u^4 (1-u)^1 = 5u^4 \, (1-u)$$
$$B_{5,5}(u) = C_{5,5} \, u^5 (1-u)^0 = u^5$$

$$C_{5,1} = \frac{\lfloor 5}{\lfloor 1 \, \lfloor 4} = \frac{5 \times \lfloor 4}{\lfloor 1 \, \lfloor 4} = 5$$

$$C_{5,2} = \frac{\lfloor 5}{\lfloor 2 \, \lfloor 3} = \frac{5 \times \lfloor 4}{\lfloor 2 \, \lfloor 3} = 5 \times 2 = 10$$

$$C_{5,3} = \frac{\lfloor 5}{\lfloor 3 \, \lfloor 2} = 10$$

$$C_{5,4} = \frac{\lfloor 5}{\lfloor 4 \, \lfloor 1} = 5$$

at $u = 0.5$

$$\begin{aligned} X(0.5) &= 0 + 1 \times 5(0.5) \, (0.5)^4 + 2 \times 10 \times (0.5)^2 \, (0.5)^3 + 3 \times \\ &\quad 10(0.5)^5 + 4 \times 5(0.5)^5 + 5 \times (0.5)^5 \\ &= (0.5)^5 \, (5 + 20 + 30 + 25) = 2.5 \end{aligned}$$

$$\begin{aligned} Y(0.5) &= 0 + 2^5 \times (0.5)^5 + 1 \times 10(0.5)^5 + 1 \times 10(0.5)^5 \\ &\quad + 10 \times 5(0.5)^5 + 5 \times (0.5)^5 \\ &= (0.5)^5 \, (10 + 10 + 10 + 50 + 5) \\ &= 85 \times (0.5)^5 = 2.6562 \end{aligned}$$

at $u = 0.5$, $(X, Y) = (2.5, 2.6562)$.

**c. Explain various types of quadratic surfaces with their equation and blobby objects.**

**Ans.** **Various types of quadratic surfaces :**

**1. Sphere :**

a. A sphere with a radius $r$ is defined as the set of points $(x, y, z)$ that satisfies the following equations :

$$x^2 + y^2 + z^2 = r^2$$
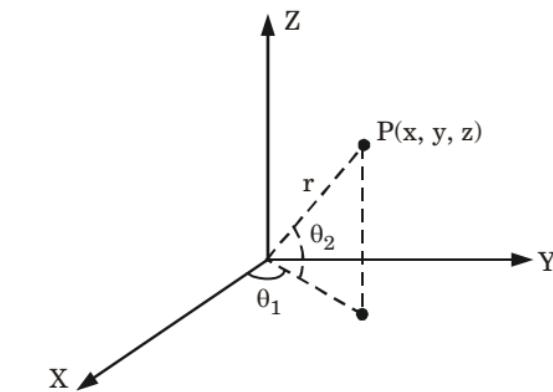
b. The parametric equations for spherical surfaces are :

$$x = r \cos \theta_1 \cos \theta_2$$
$$y = r \sin \theta_1 \cos \theta_2$$
$$z = r \sin \theta_2$$
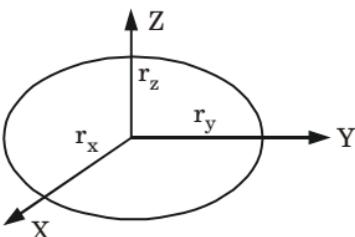
where $-\dfrac{\pi}{2} \le \theta_2 \le \dfrac{\pi}{2}$ and $-\pi \le \theta_1 \le \pi$

**Fig. 5.** Parametric coordinate position $(r, \theta_1, \theta_2)$
on the surface of a sphere.

**2. Ellipsoid :**

a. An ellipsoid surface is the extension of a spherical surface. There are three mutually perpendicular radii that have different values as shown in Fig. 6.



**Fig. 6.** An ellipsoid with three mutually perpendicular
radii $r_x$, $r_y$ and $r_z$, centred on the coordinate origin.

b. In the Cartesian coordinate system, the ellipsoid centered at the origin can be defined as :

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

c. The parametric equations are :

$$x = r_x \cos \theta_1 \cos \theta_2$$
$$y = r_y \sin \theta_1 \cos \theta_2$$
$$z = r_z \sin \theta_2$$
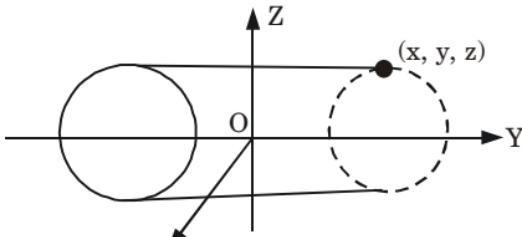
where, $-\pi/2 \leq \theta_1 \leq \pi/2$ and $-\pi \leq \theta_2 \leq \pi$

**3. Torus :**

a. A doughnut-shaped object as shown in Fig. 7 is called a torus.



**Fig. 7.** Torus

This can be generated by rotating a conic object like a circle about a specified axis.

b. In cartesian coordinates, equations for points over the surface of a torus is given as

$$\left[ r - \sqrt{\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2} \right]^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

c. The parametric representations for a torus can be given as

$$x = r_x(r + \cos\phi)\cos\theta, \qquad -\pi \le \phi \le \pi$$
$$y = r_y(r + \cos\phi)\sin\theta, \qquad -\pi \le \theta \le \pi$$
$$z = r_z\sin\phi$$

**Blobby objects :**

1. Blobby objects do not maintain a fixed shape and change their surface characteristics in certain motions or when they come in contact with other objects.
2. Molecular structures, melting objects, water droplets, muscle shapes in the human body, etc., are examples of blobby objects.
3. As objects show a certain degree of fluidity and blobbiness, these objects are simply known as blobby objects.



**Fig. 8.** Molecular bonding as two molecules move away from each other, the surface shape stretch, snap and finally contract into spheres.



**Fig. 9.** Blobby muscle shape in the human arm.

4. Answer any **two** parts of the following :                    **(7 × 2 = 14)**
a. **What do you understand by the term "Concatenation of transformations" ? What are its advantages ? If $A$ and $B$ are two different transformations, illustrate with suitable example that $A \cdot B \ne B \cdot A$.**

Ans.  **Concatenation of transformation :**
1. Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.
2. A combined matrix is given by :
    $$[T][X] = [X][T_1][T_2][T_3][T_4]....[T_n]$$
3. A composite transformation is two or more transformations performed one after the other.
4. If a transformation of the plane $T_1$ is followed by a second plane transformation $T_2$, then the result itself may be represented by a single transformation $T$ which is the composition of $T_1$ and $T_2$ taken in that order and written as :
    $$T = T_1.T_2$$

5. The change in the order of transformation would lead to different results, as in general matrix multiplication is not commutative, *i.e.*, [A]. [B] ≠ [B]. [A] and the order of multiplication.

**Advantages of concatenation of transformation :**

More complex geometric and coordinate transformation are formed through process of concatenation of matrices.

**For example,**

$$A = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h & k & 1 \end{pmatrix}$$
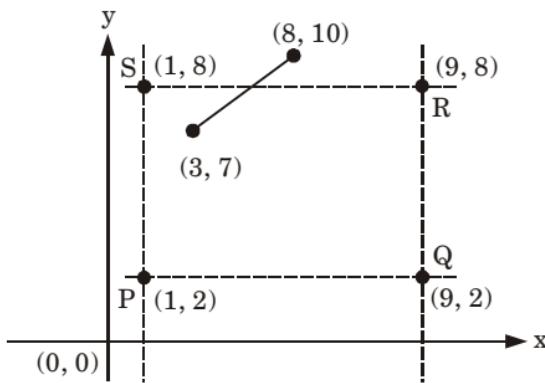
$$A \cdot B = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h & k & 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ h & k & 1 \end{pmatrix}$$

$$B \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h & k & 1 \end{pmatrix} \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ hs & ks & 1 \end{pmatrix}$$

$$A \cdot B \neq B \cdot A$$

**b. Apply the Liang-Barsky to clip the line segment from A (3,7) to B (8,10) against the regular rectangular window P (1,2), Q (9,2), R (9,8) and S (1, 8).**

**Ans.**



**Fig. 10.**

Line segments points are (3, 7) and (8, 10).

Rectangular window points are $P$ (1, 2), $Q$ (9, 2), $R$ (9, 8) and S (1, 8).

$$x_{max} = 9 \quad y_{max} = 8$$
$$x_{min} = 1 \quad y_{min} = 2$$
$$(x_1, y_1) = (3, 7)$$
$$(x_2, y_2) = (8, 10)$$
$$\Delta x = x_2 - x_1 = 8 - 3 = 5$$
$$\Delta y = y_2 - y_1 = 10 - 7 = 3$$

$$p_1 = -\Delta x = -5 \qquad\qquad q_1 = x_1 - x_{min} = 3 - 1 = 2$$
$$p_2 = \Delta x = 5 \qquad\qquad q_2 = x_{max} - x_1 = 9 - 3 = 6$$
$$p_3 = -\Delta y = -3 \qquad\qquad q_3 = y_1 - y_{min} = 7 - 2 = 5$$
$$p_4 = \Delta y = 3 \qquad\qquad q_4 = y_{max} - y_1 = 8 - 7 = 1$$

$$\mu_1 = r_1 = q_1 \,|\, p_1 = \frac{2}{-5} = -0.4$$

$$\mu_2 = r_2 = q_2 \,|\, p_2 = \frac{6}{5} = 1.2$$

$$\mu_3 = r_3 = q_3 \,|\, p_3 = \frac{5}{-3} = -1.66$$

$$\mu_4 = r_4 = q_4 \,|\, p_4 = \frac{1}{3} = 0.33$$

$$\mu_1 = \max(-0.4, -1.66, 0) = 0$$
$$\mu_2 = \min(1, 1.2, 0.33) = 0.33$$
$$\mu_1 < \mu_2 \text{ endpoints are visible}$$
$$x'_1 = x_1 + (\Delta x \times \mu_1) = 3 + (5 \times 0) = 3$$
$$y'_1 = y_1 + (\Delta y \times \mu_1) = 7 + (3 \times 0) = 7$$
$$x'_2 = x_1 + (\Delta x \times \mu_2) = 3 + (5 \times 0.33) = 4.65$$
$$y'_2 = y_1 + (\Delta y \times \mu_2) = 7 + (3 \times 0.33) = 7.99$$

$\therefore$  Visible line will be $P_1(3, 7)$ and $P_2(4.65, 7.99)$.

**c. Explain midpoint circle algorithm.**

**Ans.** **Midpoint circle algorithm :**
1. Input radius $r$ and center $(x_c, y_c)$ and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$ *i.e.*, initialize starting position as
$$x = 0$$
$$y = r$$
2. Calculate the initial value of the decision parameter as
$$p = 1.25 - r$$
3. do
```
        {
            plot (x, y)
            if (p < 0)
                {
                  x = x + 1
                  y = y
                  p = p + 2x + 1
                }
            else
                {
                  x = x + 1
                  y = y - 1
```

$$p = p + 2x - 2y + 1$$
$$\}$$
$$\}$$

while $(x < y)$

4. Determine symmetry points in the other seven octants.
5. Stop.

☺☺☺

# B.Tech.
# (SEM. IV) EVEN SEMESTER THEORY
# EXAMINATION, 2015-16
# COMPUTER GRAPHICS

**Time : 3 Hours**                    **Max. Marks : 100**

**Note :** Attempt questions from **all** sections as per directions.

### SECTION - A

1. Attempt **all** parts of this section. Answer in brief.    **(2 × 10 = 20)**
a. **Explain the role of pixel and frame buffer in graphics devices.**

b. **What do you understand by the video controller ?**

c. **Justify the composite transformation.**

d. **What is specular reflection ?**

e. **Write the properties of B-Spline curve.**

f. **What do you understand by the raster scan display ?**

g. **Write the rotation matrix about $x$, $y$ and $z$-axis in 3D.**

h. **Write the Bresenham's algorithm of a line.**

i. **Give window to viewport transformation matrix.**

j. **What is the persistence of phosphor ?**

### SECTION-B

2. Attempt any **five** questions from this section.    **(10 × 5 = 50)**
a. **Compare the computation done in Digital Differential Analyzer (DDA) algorithm with Bresenham's line drawing algorithm.**

b. **Write a procedure for rotation. Derive reflection metrics for reflection about $X$-axis.**

c. **Consider two raster systems with resolutions of 640 × 480 and 1280 × 1024. How many pixels could be accessed per second in each of these systems by a display controller that refreshes the screen at a rate of 60 frames per second ?**

d. **Write an algorithm for Cohen-Sutherland line clipping algorithm. Compare it with Liang-Barsky line clipping algorithm.**

e. **What is window to view point coordinate transformation ? What are the issues related to multiple windowing ?**

f. **Explain parallel and perspective projection. Justify the depth cueing projection for 3D display methods.**

g. **What are the criteria that should be satisfied by a good line drawing algorithm ? Explain in detail.**

h. **Explain the mid-point circle generation algorithm.**

#### SECTION-C

Attempt any **two** questions from this section.          **(15 × 2 = 30)**

3. **List the advantages and disadvantages of back-face detection and A-buffer method. Write the algorithm for back-face detection.**

4. **Explain the different illumination methods and different rendering methods in detail. Write its advantages and disadvantages.**

5. **Compare and contrast among spline, B-spline and Bezier algorithms for curve generation and write the algorithm for Bezier curve generation.**

☺☺☺

## **SOLUTION OF PAPER (2015-16)**

**Note :** Attempt questions from **all** sections as per directions.

### **SECTION - A**

**1.** Attempt **all** parts of this section. Answer in brief.       **(2 × 10 = 20)**

**a. Explain the role of pixel and frame buffer in graphics devices.**

**Ans.** **Role of pixel :** A pixel is a smallest size object or colour spot that can be displayed or addressed on a monitor. They represent the smallest controllable element of a picture on the screen.
**Role of frame buffer :** Picture definition is stored in a memory area called the refresh buffer or frame buffer. It holds the set of intensity values for all the screen points.

**b. What do you understand by the video controller ?**

**Ans.** Video controller is one of the components of an interactive raster scan system. It is used to control the operation and intensity information of each pixel from frame buffer and displays them on screen.

**c. Justify the composite transformation.**

**Ans.**
1. A composite transformation is two or more transformations performed one after the other.
2. If a transformation of the plane $T_1$ is followed by a second plane transformation $T_2$, then the result itself may be represented by a single transformation $T$ which is the composition of $T_1$ and $T_2$ taken in that order.
3. This is written as $T = T_1.T_2$.
4. Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.

**d. What is specular reflection ?**

**Ans.** Specular reflection is the phenomenon of reflection of incident light in a concentrated region around the specular reflection angle.

**e. Write the properties of B-Spline curve.**

**Ans.** **Properties of B-spline curves are :**
   i. The polynomial curve has degree $d-1$ and $C^{d-2}$ continuity over the range of 4.
  ii. For $n+1$ control points, the curve is described with $n+1$ blending functions.

**f. What do you understand by the raster scan display ?**

Ans. Raster scan display is a method of displaying images on the CRT screen. In this, the horizontal and vertical deflection signals are generated to move the beam all over the screen in a pattern.

**g. Write the rotation matrix about $x$, $y$ and $z$-axis in 3D.**

Ans.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**h. Write the Bresenham's algorithm of a line.**

Ans. **Bresenham's line drawing algorithm :**

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

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

3. Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for decision parameter as :

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

4. At each $x_k$ along the line, starting at $k = 0$, perform the following test :

   a.  If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and

       $$p_{k+1} = p_k + 2\Delta y$$
   b.  If $p_k > 0$, the next point to plot is $(x_k + 1, y_k + 1)$ and
       $$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$
5. Repeat step 4 $\Delta x$ times.

**i. Give window to viewport transformation matrix.**

Ans.

$$\begin{bmatrix} S_x = \dfrac{V_{x\max} - V_{x\min}}{W_{x\max} - W_{x\min}} & 0 & 0 \\ 0 & S_x = \dfrac{V_{y\max} - V_{y\min}}{W_{y\max} - W_{y\min}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**j. What is the persistence of phosphor ?**

**Ans.** Persistence is defined as the time taken by the emitted light from the screen to decay to one-tenth of its original intensity. A phosphor with low persistence is useful for animation; a high persistence phosphor is useful for displaying highly complex, static pictures.

**SECTION-B**

**2.** Attempt any **five** questions from this section. **(10 × 5 = 50)**

**a. Compare the computation done in digital differential analyzer (DDA) algorithm with Bresenham's line drawing algorithm.**

**Ans.**

| S. No. | Digital Differential Analyzer (DDA) line drawing | Bresenham's line drawing Algorithm |
|--------|--------------------------------------------------|-------------------------------------|
| 1. | DDA algorithm uses floating points *i.e.*, real arithmetic. | Bresenham's algorithm uses fixed points *i.e.*, integer arithmetic. |
| 2. | DDA algorithm uses multiplication and division in its operations. | Bresenham's algorithm uses subtraction and addition in its operations. |
| 3. | DDA algorithm is slowly than Bresenham's algorithm in line drawing. | Bresenham's algorithm is faster than DDA algorithm in line drawing. |
| 4. | DDA algorithm round off the coordinates to integer that is nearest to the line. | Bresenham's algorithm does not round off but takes the incremental value in its operation. |

**b. Write a procedure for rotation. Derive reflection metrics for reflection about X axis.**

**Ans.** **Procedure for rotation :** Rotation can be done in the following order :

**Step 1 :** Translate the object or body at the origin (translation).

**Step 2 :** Rotate by any angle as given (rotation).

**Step 3 :** Translate back to its original location (inverse translation).

In matrix form, it can be shown as

$$[T] = [T_R] \ [R_\theta] \ [T_R]^{-1}$$

where $T_R \rightarrow$ Translation matrix

$R_\theta \rightarrow$ Rotation matrix by an angle θ°

$[T_R]^{-1} \rightarrow$ Inverse translation matrix

**Reflection metrics :** For reflection about $x$-axis, $x$ coordinate is not changed and sign of $y$ coordinate is changed. Thus if we reflect point $(x, y)$ in the $x$-axis, we get $(x, -y)$

$i.e.$,             $x' = x$

$y' = -y$

So, the transformation matrix for reflection about $x$-axis or $y = 0$ axis is,

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

and the transformation is represented as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

**c. Consider two raster systems with resolutions of 640 × 480 and 1280 × 1024. How many pixels could be accessed per second in each of these systems by a display controller that refreshes the screen at a rate of 60 frames per second ?**

**Ans.** Number of pixel in our frame (resolution) = Number of column (width) × number of row (height)

Number of frame in one second = 60 frames

So, number of pixel in 60 frames = resolution × 60

Hence, for resolution 640 × 480,

Number of pixels = 640 × 480 × 60

= 18432000 pixels
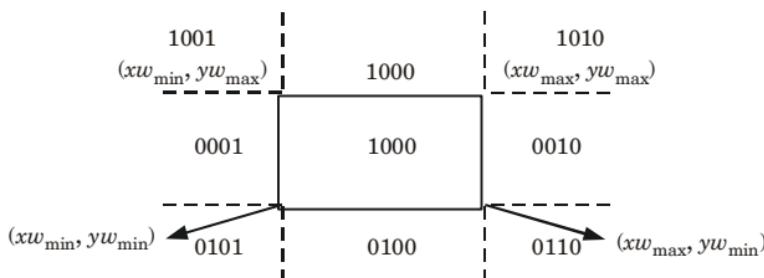
For resolution 1280 × 1024,

Number of pixels = 1280 × 1024 × 60

= 78643200 pixels

**d. Write an algorithm for Cohen-Sutherland line clipping algorithm. Compare it with Liang-Barsky line clipping algorithm.**

**Ans.** **Cohen-Sutherland algorithm :**

1. In this algorithm, a rectangular window is considered whose coordinates are $(xw_{min}, yw_{min}, xw_{max}, yw_{max})$ as shown in Fig. 1.



**Fig. 1.**

2. Every line end point is assigned a 4-digit binary code (region code) that identifies the location of the point corresponding to the boundaries of the clipping window.

3. The procedure for calculating the end point $p(x, y)$ code is as follow :

   **a. For bit 1**
   If $(y - yw_{max}) < 0$ then
   code = 0;
   else
   code = 1;

   **b. For bit 2**
   If $(yw_{min} - y) < 0$ then
   code = 0;
   else
   code = 1;

   **c. For bit 3**
   if $(x - xw_{max}) < 0$ then
   code = 0;
   else
   code = 1;

   **d. For bit 4**
   if $(xw_{min} - x) < 0$ then
   code = 0;
   else
   code = 1;

4. If the region code of both the end points of a line segment is 0000, the line is completely visible as it completely lies within the window.

5. If the region code of both the end points of the line segment is not 0000 then we calculate AND operation of both end points *A* and *B*.

6. If the result of AND operation is 0000, the line is partially visible and then find the point of intersection.

   **Comparison :**
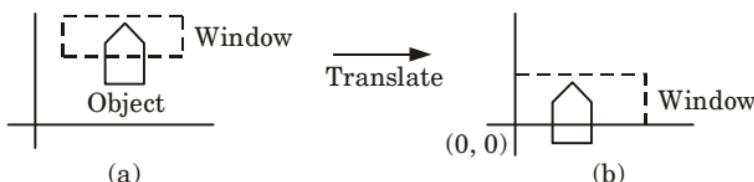
| S. No. | Cohen Sutherland line clipping algorithm | Liang-Barsky line clipping algorithm |
|--------|------------------------------------------|--------------------------------------|
| 1. | It is not the most efficient but efficient when most of the lines to be clipped are either rejected or accepted. | It is more efficient than Cohen Sutherland since intersection calculations are reduced. |
| 2. | Each intersection calculation requires both multiplication and a division. | Each update of parameters requires only one division. |
| 3. | It repeatedly calculates intersection along a line path even through the line may be completely outside the clip window. | Window intersections of lines are calculated once when final values have been computed. |

**e. What is window to view point coordinate transformation ? What are the issues related to multiple windowing ?**

**Ans.** **Window to view point coordinate transformation :**
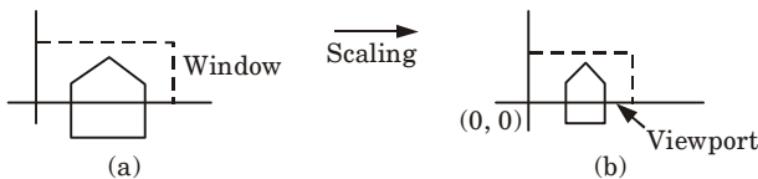
1. Window to viewport / view point Transformation is the process of transforming a 2D world-coordinate objects to device coordinates.

2. Objects inside the world or clipping window are mapped to the viewport which is the area on the screen where world coordinates are mapped to be displayed.

3. Window to viewport mapping or transformation is done in following three steps :

   **Step 1 :** The object together with its window is translated until the lower-left corner of the window is at the origin.
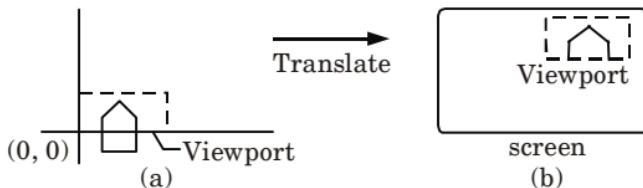


**Fig. 2.**

**Step 2 :** The object and the window are then scaled until the window has the dimension same as viewport. In other words, we are converting the object into image and window in viewport.



**Fig. 3.**

**Step 3 :** The final transformation step is another translation to move the viewport to its correct position on the screen.



**Fig. 4.**

**Issues related to multiple windowing :**

1. A key problem in multi-windowing is how to automatically assign windows to the relevant areas within an image.

2. More difficult to work with several applications at once.

**f. Explain parallel and perspective projection. Justify the depth curing projection for 3D display methods.**

**Ans.**
1. **Parallel projection :**
a. In a parallel projection, coordinate positions are transformed to the view plane along parallel lines.
b. These are linear transforms that are useful in blueprints to produce scale drawings of three dimensional objects.
2. **Perspective projection :**
a. In a perspective projection, object positions are transformed to the view plane along lines that converge to a point called the center of projection.
b. The projected view of an object is determined by calculating the intersection of the projection lines with the view plane.
   **Depth cueing projection :**
1. In depth cueing, we include the depth information in the displayed object.
2. The depth of an object can be represented by the intensity of the image.
3. Depth cueing is applied by choosing maximum and minimum intensity values.
4. The parts of the objects closest to the viewing position are displayed with the highest intensities and objects farther away are displayed with decreasing intensities.
5. Reason to use depth cueing projection for 3D display method:
a. To create realistic view of an object.
b. To easily identify the particular viewing direction of displayed object.
6. A simple method for indicating depth is using wireframe display.
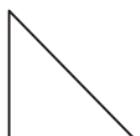
**g. What are the criteria that should be satisfied by a good line drawing algorithm ? Explain in detail.**

**Ans.** **Criteria for good line drawing algorithm :**
1. **Lines should appear straight :**
a. Lines generated parallel to the *x*-axis or *y*-axis or at 45° are plotted correctly with point plotting techniques.
b. Line may not pass through other addressable points in between.
2. **Lines should terminate accurately :**
a. If lines are not plotted accurately, they may terminate at wrong places.
b. This may lead to a small gap between endpoints of one line and the starting point of the next as shown in Fig. 5.



**Fig. 5.**

3. **Line should have constant density :**
a. To maintain the constant density throughout the line, dots should be equally spaced.

b. The line density is proportional to the number of dots displayed divided by the length of the line.

c. Dots are equally spaced only in lines parallel to the $x$-axis or $y$-axis or at 45° to the axis as given in Fig. 6.



**Fig. 6.**

4. **Line brightness independent of slope and line length :** This requires a high resolution of the device along with a high refresh rate.

5. **Lines should be drawn rapidly :** In interactive applications, lines should appear rapidly on the screen, that is, minimum computation is desired to draw the line.

**h. Explain the midpoint circle generation algorithm.**

**Ans.** **Mid-point circle algorithm :**

1. Input radius $r$ and center $(x_c, y_c)$ and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$ *i.e.*, initialize starting position as

$$x = 0$$
$$y = r$$

2. Calculate the initial value of the decision parameter as

$$p = 1.25 - r$$

3. do
   {
   plot $(x, y)$
   if $(p < 0)$
   {
   $$x = x + 1$$
   $$y = y$$
   $$p = p + 2x + 1$$
   }
   else
   {
   $$x = x + 1$$
   $$y = y - 1$$
   $$p = p + 2x - 2y + 1$$
   }
   }
   while $(x < y)$

4. Determine symmetry points in the other seven octants.

5. Stop.

## SECTION-C

Attempt any **two** questions from this section.  **(15 × 2 = 30)**

**3. List the advantages and disadvantages of back-face detection and A-buffer method. Write the algorithm for back-face detection.**

**Ans.** **Advantages of back-face detection :**

1. It is fast.
2. It is a simple object-space method.

**Disadvantages of back-face detection :**

1. It can only be used on solid objects modeled as a polygon mesh. This is the most general modeling construct for scan line graphics systems.
2. It works fine for convex polyhedra but not necessarily for concave polyhedra.

**Advantages of A-buffer method :**

i. It is an extension of the ideas in the depth-buffer method.
ii. More than one surface intensity can be taken into consideration at each pixel position.
iii. It also antialiased the object edges.
iv. It has two fields–depth field and intensity field.
v. In this, surfaces are sub-divided into a polygon mush and clipped against the pixel boundaries.
vi. The intensity of pixel determined by considering opacity parameter and percentage of overlaps of the overlapping surface.

**Disadvantages of A-buffer method :**

i. Since the algorithm processes multiple objects at a time, the total number of polygons in a picture can be arbitrarily large.

**Back-face detection algorithm :**

1. Object surfaces that are orientated away from the viewer are called back-faces.
2. Algorithm used to detect the back-faces is known as back-face detection algorithm.
3. If any three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$ on any plane surface are known, the unknown parameters $A$, $B$, $C$ and $D$ of the plane surface equation $Ax + By + Cz + D = 0$ can be found as follows :

i. All the three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$ should satisfy the equation $Ax + By + Cz + D = 0$ as it lies on the surface.
 Hence, $Ax_1 + By_1 + Cz_1 + D = 0$
     $Ax_2 + By_2 + Cz_2 + D = 0$
     $Ax_3 + By_3 + Cz_3 + D = 0$

ii. Also, any arbitrary point $(x, y, z)$ lying on the surface should satisfy the equation of the desired surface

$$Ax + By + Cz + D = 0$$

iii. A unique solution of equations for $A$, $B$, $C$ and $D$ can only be obtained if

or $\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x & y & z & 1 \end{vmatrix} = 0$ (By Cramer's rule)

or $\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 & 0 \\ x_1-x_2 & y_1-y_2 & z_1-z_2 & 0 \\ x_2-x_3 & y_2-y_3 & z_2-z_3 & 0 \\ x & y & z & 1 \end{vmatrix} = 0$

or $\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 \\ x_1-x_2 & y_1-y_2 & z_1-z_2 \\ x_2-x_3 & y_2-y_3 & z_2-z_3 \end{vmatrix} = 0$

or $(x-x_1)\begin{vmatrix} y_1-y_2 & z_1-z_2 \\ y_2-y_3 & z_2-z_3 \end{vmatrix} - (y-y_1)\begin{vmatrix} x_1-x_2 & z_1-z_2 \\ x_2-x_3 & z_2-z_3 \end{vmatrix}$
$$+ (z-z_1)\begin{vmatrix} x_1-x_2 & y_1-y_2 \\ x_2-x_3 & y_2-y_3 \end{vmatrix} = 0$$

or $\begin{vmatrix} y_1-y_2 & z_1-z_2 \\ y_2-y_3 & z_2-z_3 \end{vmatrix} x + \begin{vmatrix} z_1-z_2 & x_1-x_2 \\ z_1-z_3 & x_2-x_3 \end{vmatrix} y$
$$+ \begin{vmatrix} x_1-x_2 & y_1-y_2 \\ x_2-x_3 & y_2-y_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0$$

which is in the form of $Ax + By + Cz + D = 0$, giving the equation of the plane passing through the three points

where,    $A = \begin{vmatrix} y_1-y_2 & z_1-z_2 \\ y_2-y_3 & z_2-z_3 \end{vmatrix}$

$B = \begin{vmatrix} z_1-z_2 & x_1-x_2 \\ z_2-z_3 & x_2-x_3 \end{vmatrix}$

$C = \begin{vmatrix} x_1-x_2 & y_1-y_2 \\ x_2-x_3 & y_2-y_3 \end{vmatrix}$

and    $D = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$

iv. Any plane with its inside and outside surfaces can be defined by three points and its normal vector.

v. This normal vector can be calculated with three position vectors of three points $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$ and $(x_3, y_3, z_3)$.

vi. Using right hand thumb rule, the three points are considered in counter clockwise manner for calculating the unit normal directing

outside the surface. Thus, the surface normal and equation of the plane are calculated.

vii. Now, if any point $(x, y, \text{or } z)$ lies inside the polygon surface, $Ax + By + Cz + D < 0$, and if it is along the line of sight to the surface, the polygon must be back-face.

viii. If $L$ is the line of sight or viewing vector (light vector in case of shadowing), perpendicular to the viewing plane originating from the eye or camera, $N$ is the unit surface normal vector as shown in Fig. 7(a) and (b) the polygon is the back-face if $L.N > 0$
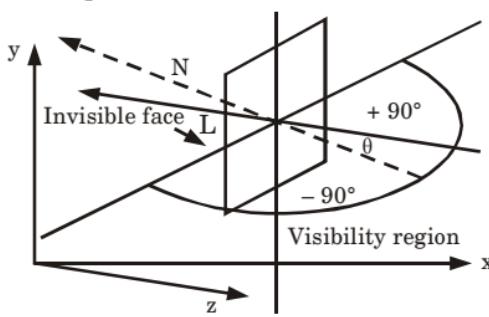


**Fig. 7.** (a) Geometry of camera vision.

ix. On aligning the light vector to that of the view vector which is normally along negative $z$-axis for right-handed viewing system, that is

$$L = 0i + 0j + L_z k$$
$$L.N = L_z K$$

x. Hence, only $z$-component of normal vector $N$ is required to be considered, and the sign of $K$ is checked for negative value.

xi. Thus, any polygon is a back-face if it has a surface normal with negative $z$-component value.



**Fig. 7.** (b) Invisible and visible vision.

**4. Explain the different illumination methods and different rendering methods in detail. Write its advantages and disadvantages.**

**Ans.**   **Various illumination models :**

**1. Ambient light :**

i. In this model, the intensity of the reflected light for each surface depends on the optical properties of the surface.

ii. Ambient light has no spatial or directional characteristics.

  iii. The amount of ambient light incident on each object surface is constant for all surfaces and over all directions.

  iv. The optical properties determine how much of the incident energy is to be reflected and how much absorbed.

**2. Diffuse reflection :**

  i. In this model, light coming from all directions is reflected from the walls, floor, ceiling, etc.

  ii. Diffuse reflections are constant over each surface in a scene, independent of the viewing direction.

  iii. The amount of diffused reflected light for each surface in a scene can be set with parameter $K_d$.

  iv. For the surfaces which absorbs most of the incident light (black surfaces), the value of the diffuse reflection coefficient ($K_d$) is close to 0.

  v. If a surface is exposed to only ambient light, then the intensity of the diffuse reflection at any point on the surface is given by

$$I_{ambdiffuse} = K_d I_a$$



(a) Radiant energy from surface.

(b) Diffuse reflection from surface.

**Fig. 8.**

**3. Phong model :**

  i. According to the Phong model, the intensity of specular reflection is proportional to $\cos^{n_s} \theta$. The value of $n$, specular reflection parameter, is determined by the type of surface.

  ii. For very shiny surfaces, the value of $n_s$ is large (more than 100) and for dull surfaces the value of $n_s$ is close to 1. For a perfect reflector, the value of $n_s$ is infinite.

  iii. The intensity of specular reflection depends on the angle of incidence, on the properties of the surface, colour of incident light and polarization.

  iv. The specular reflection coefficient $W(\theta)$, for each surface may be used to model the monochromatic specular intensity variation.

  v. We can write the Phong specular reflection model as

$$I_{spec} = W(\theta) I_l \cos^{n_s} \theta \qquad \qquad ...(1)$$

where $I_l$ is intensity of light source, and $\theta$ is the viewing angle relative to specular reflection direction $R$.

  vi. In the case of opaque materials, the specular reflection rejection is almost constant for all incidence angles. Therefore, replacing $W(\theta)$ with a constant specular reflection coefficient $K_s$. Since $V$ and $R$ are

unit vectors in the viewing and specular reflection directions, $\cos \theta = V.R$. We can rewrite the equation (1) as

$$I_{\text{spec}} = K_s I_l (V.R)^{n_s}$$



**Fig. 9.** Projection of L on to the direction of normal vector N.

**Advantages of illumination methods :**
1. Fast
2. Acceptable results
3. Hardware support

**Disadvantages of illumination methods :**
1. Point light source
2. No interaction between objects
3. Adhoc, not based on model of light propagation

**Different rendering methods :**

**1. Constant-intensity shading :**
a. In this method, a single intensity is calculated for each polygon and all points in the surface of polygon are displayed with the same intensity value.
b. It is fast and simplest rendering method.
c. Light sources illuminating the object and the viewing position is sufficiently far from the surface of polygon.

**2. Gouraud shading :**
a. It represents a polygon surface by linearly interpolating intensity across the polygon surface.
b. Intensity values for each polygon are matched with adjacent polygon along common edges.
c. It eliminates the intensity discontinuity that can occur in constant-intensity shading.

**3. Phong shading :**
a. Phong shading is method which interpolates normal vectors and then apply the illumination model to each surface point.
b. This method interpolates the surface normal vector, instead of the intensity.
c. It is used in 3D system.

**4. Fast Phong shading :**
a. Fast Phong shading approximates the intensity calculations using a Taylor-series expansion and triangular surface patches.
b. Surface rendering with Phong shading can be speeded up by using approximations in the illumination model calculations of normal vectors.

**Advantages of rendering methods :**

i. It removes the intensity discontinuity.

ii. It can be combined with hidden surface algorithm to fill in the visible polygons along each scanline.

iii. It displays more realistic highlights on a surface.

iv. It gives more accurate results.

**Disadvantages of rendering methods :**

i. It has a problem with specular reflections.

ii. It introduces anomalies known as Mach bands.

iii. It requires more calculations and increases the cost of shading steeply.

**5. Compare and contrast among spline, B-spline and Bezier algorithms for curve generation and write the algorithm for Bezier curve generation.**

**Ans.** **Algorithm for Bezier curve :**

1. Get four control points say A $(x_A, y_A)$, B$(x_B, y_B)$, C$(x_C, y_C)$, D$(x_D, y_D)$.

2. Divide the curve represented by points $A$, $B$, $C$ and $D$ in two sections

$$x_{AB} = (x_A + x_B) / 2$$
$$y_{AB} = (y_A + y_B) / 2$$
$$x_{BC} = (x_B + x_C) / 2$$
$$y_{BC} = (y_B + y_C) / 2$$
$$x_{CD} = (x_C + x_D) / 2$$
$$y_{CD} = (y_C + y_D) / 2$$
$$x_{ABC} = (x_{AB} + x_{BC}) / 2$$
$$y_{ABC} = (y_{AB} + y_{BC}) / 2$$
$$x_{BCD} = (x_{BC} + x_{CD}) / 2$$
$$y_{BCD} = (y_{BC} + y_{CD}) / 2$$
$$x_{ABCD} = (x_{ABC} + x_{BCD}) / 2$$
$$y_{ABCD} = (y_{ABC} + y_{BCD}) / 2$$

3. Repeat the step 2 for section $A$, $AB$, $ABC$ and $ABCD$ and section *ABCD, BCD*, *CD* and *D*.

4. Repeat step 3 until we have sections so short that they can be replaced by straight lines.

5. Replace small sections by straight lines.

6. Stop.

**Comparison :**

| S. No. | Spline | B-spline | Bezier |
|--------|--------|----------|--------|
| 1. | A spline curve can be specified by giving a set of coordinate positions, called control points which indicates the general shape of the curve. | The B-spline curves are specified by a Bernstein basis function that has a limited flexibility. | The Bezier curves can be specified with boundary conditions, with a characterizing matrix or with blending functions. |
| 2. | The curve follows the general shape of the curve. | The curves resulted by the use of open uniform basis function are nearly like Bezier curves. | The curve generally follows the shape of defining polygon. |
| 3. | Typical CAD application for splines include the design of automobile bodies, aircraft and space craft surfaces and ship hulls. | These curves can be used to construct blending curves. | Bezier curves are found in painting & drawing packages as well as CAD system. |
| 4. | It possess a high degree of smoothness at the places where the polynomial pieces connect. | The B-spline allows the order of the basis function and hence the degree of the resulting curve is independent on the number of vertices. | The degree of the polynomial defining the curve segment is one less than the number of defining polygon point. |

☺☺☺

# B. Tech.
# (SEM. VI) EVEN SEMESTER THEORY
# EXAMINATION, 2018-19
# COMPUTER GRAPHICS

**Time : 3 Hours** **Max. Marks : 70**

**Note :** Attempt **all** Section. If require any missing data; then choose suitable.

## SECTION-A

**1.** Attempt **all** questions in brief : **(2 × 7 = 14)**

**a.** What are the applications of computer graphics ?

**b.** How many clippers are used by Sutherland Hodgeman for polygon clipping ?

**c.** Define aspect ratio and types of retracing ?

**d.** What is tilting tranformation ? Does the order of performing the rotation matter ?

**e.** What do you understand by match band effect and transparency ?

**f.** Explain other transformation that can be applied on 2D objects ?

**g.** Define blobby objects and types of coherence.

## SECTION-B

**2.** Attempt any **three** of the following : **(7 × 3 = 21)**

**a.** Why do we need video controller ? Also define the architecture of raster scan system ?

**b.** Translate the square *ABCD* whose coordinates are *A*(0, 0), *B*(3, 0), *C*(3, 3), and *D*(0, 3) by 2 unit in both directions and then scale it by 1.5 units in *x*-direction and 0.5 unit in *y*-direction.

**c.** Write rotation matrices about *X*-axis, *Y*-axis and *Z*-axis and prove that for any rotation matrix $R$ :- $R^{-1}(\theta) = R(-\theta) = R^T(\theta)$

    **d. Discuss RGB and CMY colour model in detail.**

    **e. Explain the true-curve generation algorithm. Also list the problems in this algorithm.**

### SECTION-C

**3.** Attempt any **one** part of the following :               $(7 \times 1 = 7)$
**a. What are the disadvantages of DDA algorithm ? Also write Bresenham's line drawing algorithm for negative slope.**

**b. Write mid-point circle algorithm and predict the pixels in any octant of circle for radius = 12 pixels with its centre at origin ?**

**4.** Attempt any **one** part of the following :               $(7 \times 1 = 7)$
**a. Write the Liang Barsky algorithm for line clipping. Use Liang Barsky line clipping algorithm to clip the line $P_1(-1, 7)$ to $P_2(11, 1)$ against the window having diagonally opposite corners as (1, 2) and (9, 8).**

**b. Explain window-to-viewport transformation in detail.**

**5.** Attempt any **one** part of the following :               $(7 \times 1 = 7)$
**a. A rectangular parallelopiped is given having length on *X*-axis, *Y*-axis and *Z*-axis as 3, 2 and 1 respectively. First apply a rotation of – 90° about the *Y*-axis followed by rotation of 90° about *X*-axis.**



**Fig. 1.**

**b. What do you understand by projection ? Differentiate between parallel projection and perspective projection.**

**6.** Attempt any **one** part of the following :               $(7 \times 1 = 7)$
**a. Construct the Bezier curve of order 3 and with 4 polygon vertices A(1, 1), B(2, 3), C(4, 3), D(6, 4).**

**b. Write the properties of B-spline curves. Also write advantages of B-spline curves on Bezier curves.**

**7.** Attempt any **one** part of the following :               $(7 \times 1 = 7)$

a. **Explain depth buffer method and compare it with A-buffer method.**

b. **Why is Gouraud shading also referred to as interpolation shading ? Also discuss its advantages and disadvantages.**

☺☺☺

## SOLUTION OF PAPER (2018-19)

**Note :** Attempt **all** Section. If require any missing data; then choose suitable.

### SECTION-A

**1.** Attempt **all** questions in brief :         **(2 × 7 = 14)**

**a. What are the applications of computer graphics ?**

**Ans. Applications of computer graphics :**
  i. User interfaces        ii. Plotting of graphics and chart
 iii. Office automation and desktop publishing
 iv. Simulation and animation
  v. Art and commerce

**b. How many clippers are used by Sutherland Hodgeman for polygon clipping ?**

**Ans.** Four clippers are used by Sutherland Hodgeman for polygon clipping.

**c. Define aspect ratio and types of retracing ?**

**Ans. Aspect ratio :**
  1. An aspect ratio is an attribute that describes the relationship between the width and height of an image.
  2. Aspect ratio is expressed by the symbolic notation *i.e.*, $x : y$.
    **Types of retracing :** There are two types of retracing :
  1. Vertical retracing        2. Horizontal retracing

**d. What is tilting tranformation ? Does the order of performing the rotation matter ?**

**Ans.** Tilting is a transformation, where the object first rotate about $x$-axis and then about $y$-axis. We can find the required tranformation by composing two rotation matrices, *i.e.*,

$$R_{\theta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_{\theta_y} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiply $R_{\theta_y}$ and $R_{\theta_x}$, we get

$$R_{\theta_y} \cdot R_{\theta_x} = \begin{bmatrix} \cos\theta_y & \sin\theta_y\sin\theta_x & \sin\theta_y\cos\theta_x & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ -\sin\theta_y & \cos\theta_y\sin\theta_x & \cos\theta_y\cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, we multiply $R_{\theta_x}$ and $R_{\theta_y}$ we get

$$R_{\theta_x} \cdot R_{\theta_y} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ \sin\theta_x \sin\theta_y & \cos\theta_x & -\sin\theta_x \cos\theta_y & 0 \\ -\cos\theta_x \sin\theta_y & \sin\theta_x & \cos\theta_x \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, $R_{\theta} \cdot R_{\theta_y}$ is not the same matrix, as $R_{\theta_y} \cdot R_{\theta_x}$. Thus, the order of rotation matrix is important.

**e. What do you understand by match band effect and transparency ?**

**Ans.** **Mach band effect :** The linear intensity interpolation can result bright or dark intensity streaks to appear on the surface. These bright or dark intensity streaks, are called mach bands. The mach band effect can be reduced by breaking the surface into a greater number of smaller polygons.

**Transparency :**

1. Transparency is a phenomenon in which the object in the way of light ray does not obstruct light.
2. It means that if the light source is behind the object and the object is transparent, then the viewer can see the light through the object.

**f. Explain other transformation that can be applied on 2D objects ?**

**Ans.** **Basic operations performed by geometric transformation are :**

 i. Translation
 ii. Rotation
iii. Scaling

**g. Define blobby objects and types of coherence.**

**Ans.** **Blobby objects :** Blobby objects do not maintain a fixed shape and change their surface characteristics in certain motions or when they come in contact with other objects.

**Types of coherence :** There are two type of coherence :

1. Spatial coherence
2. Temporal coherence

### SECTION-B

**2.** Attempt any **three** of the following :      **(7 × 3 = 21)**
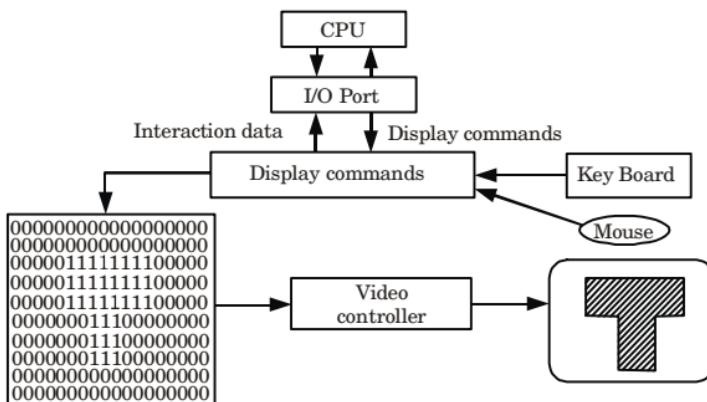
**a. Why do we need video controller ? Also define the architecture of raster scan system ?**

**Ans.** **Need of video controller :** A video controller, often referred to as a video or graphics card, is a key hardware component that

allows computers to generate graphic information to any video display devices, such as a monitor or projector.

**Architecture of raster scan system :**

1. In a raster scan system, the electron beam is swept across the screen, one row at a time from top to bottom.
2. As the electron beam moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.
3. Picture definition is stored in a memory area called the refresh buffer or frame buffer.



**Fig. 1.**

4. It holds the set of intensity values for all the screen points.
5. The stored intensity values are retrieved from frames buffer and displayed on the screen one row (scan line) at a time.
6. Each screen point is referred to as a pixel.
7. Each pixel on the screen can be specified by its row and column number. Thus by specifying row and column number we can specify the pixel position on the screen.
8. The capability of a raster scan system to store intensity information for each screen point makes it well suited for the realistic display, it consists of display controller, CPU, video controller, refresh buffer, keyboard, mouse and the CRT.

**b. Translate the square *ABCD* whose coordinates are *A*(0, 0), *B*(3, 0), *C*(3, 3), and *D*(0, 3) by 2 unit in both directions and then scale it by 1.5 units in *x*-direction and 0.5 unit in *y*-direction.**

**Ans.** Here, first operation is translate and second operation is scaling.
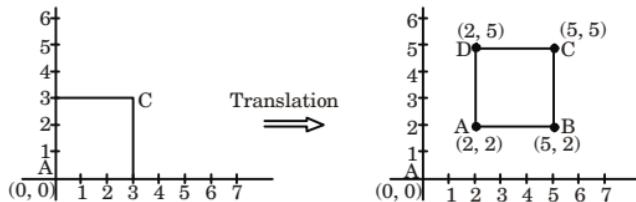
$$T_x = 2 \text{ and } T_y = 2$$

$$\text{Translation matrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

Square *ABCD* in matrix form = $\begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix}$

Performing translation operation, we get

$\begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 1 \\ 3 & 3 & 1 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 1 \\ 5 & 2 & 1 \\ 5 & 5 & 1 \\ 2 & 5 & 1 \end{bmatrix}$
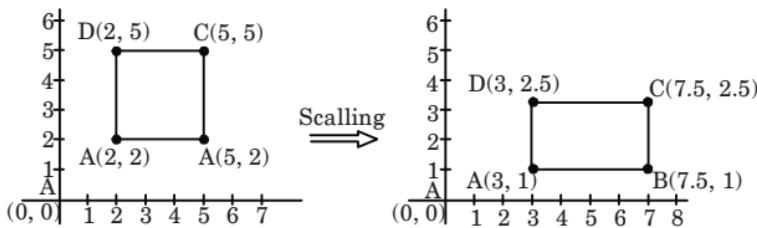


**Fig. 2.**

Now, the 2nd operation is scaling so $S_x = 1.5$, $S_y = 0.5$

and scaling matrix = $\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

After scaling we will get
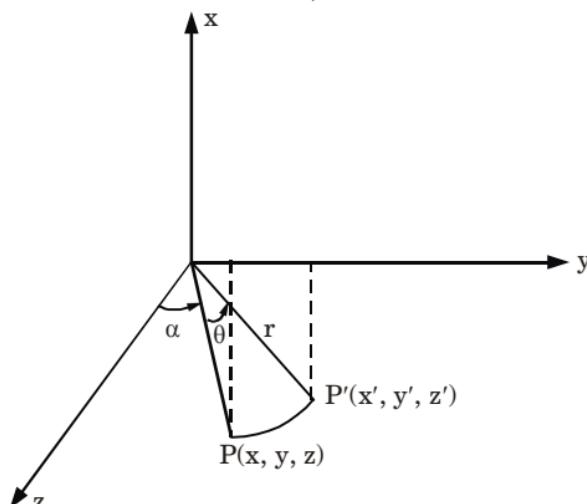
$\begin{bmatrix} 2 & 2 & 1 \\ 5 & 2 & 1 \\ 5 & 5 & 1 \\ 2 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ 7.5 & 1 & 1 \\ 7.5 & 2.5 & 1 \\ 3 & 2.5 & 1 \end{bmatrix}$



**Fig. 3.**

After transformation, new coordinates are *A*(3, 1), *B*(7.5, 1), *C*(7.5, 2.5), *D*(3, 2.5)

**c.** **Write rotation matrices about *X*-axis, *Y*-axis and *Z*-axis and prove that for any rotation matrix *R* :- $R^{-1}(\theta) = R(-\theta) = R^{T}(\theta)$**

**Ans.**

i. **Rotation about $x$-axis ($i.e.$, axis of rotation is $x$) :**



**Fig. 4.** Rotation when axis of rotation is $x$-axis.

$$x' = x$$
$$y' = r \cos (\alpha + \theta)$$
$$y' = r \cos \alpha \cos \theta - r \sin \alpha \sin \theta \qquad ...(1)$$

Since $\quad r \cos \alpha = y \quad$ and $\quad r \sin \alpha = z$

Put the value in equation (3.4.1) we get,

$$y' = y \cos \theta - z \sin \theta$$
$$z' = r \sin (\alpha + \theta)$$
$$= r \sin \alpha \cos \theta + r \cos \alpha \sin \theta$$
$$= z \cos \theta + y \sin \theta$$

or, $\qquad R_{\theta,i} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$
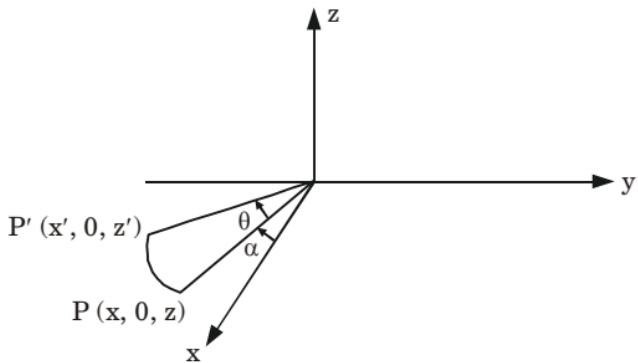
The points after rotation can be calculated by the following equation :

$$P'(x', y', z') = R_{\theta,i} P(x, y, z)$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**ii. Rotation about *y*-axis (*i.e., y*-axis is axis of rotation) :**



**Fig. 5.**

$$x' = x \cos \theta + z \sin \theta$$
$$y' = y$$
$$z' = -x \sin \theta + z \cos \theta$$
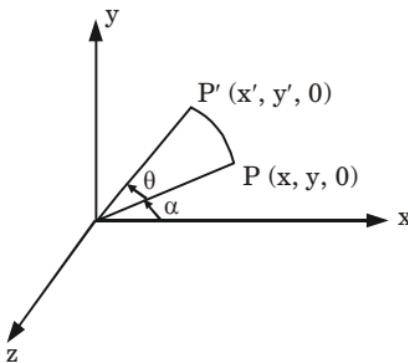
or,     $$R_{\theta, i} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**iii. Rotation about the *z*-axis (*i.e., z*-axis is axis of rotation)**



**Fig. 6.**

$$x' = x \cos \theta - y \sin \theta$$
$$y' = z \cos \theta + y \sin \theta$$
$$z' = z$$

or,     $$R_{\theta, i} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equation in the matrix form is given as :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

**Numerical :**

Let $\quad R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Put $\theta = (-\theta)$, we get

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) & 0 \\ 0 & \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since $\cos(-\theta) = \cos(\theta)$, $\sin(-\theta) = -\sin\theta$, we have

$$R(-\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now,

$$R(-\theta) * R(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

We have,

$\quad R(-\theta) . R(\theta) = I$

Multiply $R^{-1}(\theta)$ both sides,

$R(-\theta).R(\theta).R^{-1}(\theta) = I.R^{-1}(\theta)$

$\quad\quad\quad R(-\theta) = R^{-1}(\theta)$

Hence proved.

**d. Discuss RGB and CMY colour model in detail.**

**Ans.** **RGB colour model :**

1. Most graphics displays of realistic scenes are in colour. To incorporate colour, we need to write the intensity equation as a function of the colour properties of the light sources and object surfaces.

2. For an RGB description, each colour in a scene is expressed in terms of red, green, and blue components.

3. We then specify the RGB components of light source intensities and surface colours, and the illumination model calculates the RGB components of the reflected light.

4. One way to set surface colours is by specifying the reflectivity coefficients as three element vectors.

5. The diffuse reflection coefficient vector would then have RGB components $(K_{dR}, K_{dG}, K_{dB})$.

6. If we want an object to have a blue surface, we select a non-zero value in the range from 0 to 1 for the blue reflectivity component, $K_{dB}$, while the red and green reflectivity components are set to zero $(K_{dR} = K_{dG} = 0)$.

7. Any non-zero red or green components in the incident light are absorbed, and only the blue component is reflected. The intensity calculation for this example reduces to the single expression

$$I_B = K_{aB}I_{aB} + \sum_{i=1}^{n} f_i(d)I_{lBi}[K_{dB}(N.L_i) + K_{sB}(N.H_i)^{n_s}]$$

8. Surfaces typically are illuminated with white light sources, and in general we can set surface colour so that the reflected light has non-zero values for all three RGB components.

9. Calculated intensity levels for each colour component can be used to adjust the corresponding electron gun in an RGB monitor.

**CMY colour model :**

1. In CMY model, cyan, magenta and yellow are the primary colours and red, green and blue are the secondaries.

2. This model is useful for describing colour output to hard-copy devices.

3. Unlike video monitors, which procedure a colour pattern by combining light from the screen phosphors hard-copy devices such as plotters produce a colour picture by coating a paper with colour pigments.

4. Cyan is formed by adding blue and green light. When white light is reflected from cyan coloured ink, the reflected light must have no red component *i.e.*, red light is absorbed by the ink.

5. Magenta ink subtracts green components from incident light and yellow subtracts the blue components.

6. Printing process with CMY model produces collection of four ink dots, whereas RGB monitor uses a collection of three phosphor dots.

7. One dot is used for each of the primary colors (cyan, yellow and magenta) *i.e.*, one dot is black. Since the combination of 3 primary

colours produces a dark gray instead of black, so black must be included.

**e. Explain the true-curve generation algorithm. Also list the problems in this algorithm.**

**Ans.** **True-curve generation algorithm :**

1. This is the true curve generation method. Read the line endpoint $(x_1, y_1)$ and $(x_2, y_2)$

2. $$\Delta x = |x_2 - x_1|$$
   $$\Delta y = |y_2 - y_1|$$

3. If $(\Delta x \geq \Delta y)$ then

       length $= \Delta x$

   else

       length $= \Delta y$

4. Select the raster unit, $i.e.$,

   $$\Delta x = \frac{(x_2 - x_1)}{\text{length}}$$

   $$\Delta y = \frac{(y_2 - y_1)}{\text{length}}$$

5. $$x = x_1 + 0.5 * \text{Sign} (\Delta x)$$
   $$y = y_1 + 0.5 * \text{Sign} (\Delta y)$$

   Where sign function makes the algorithm work in all quadrants. It returns $-1, 0, 1$ depending on whether its agreement is $< 0, = 0, > 0$ respectively.

   The factor 0.5 makes it possible to round the values in the integer function rather than truncating them.

   Now plot the points

   $i = 1$

   while $(i \leq \text{length})$

   {

   Plot (integer $(x)$, integer $(y)$)

   $x = x + \Delta x$

   $y = y + \Delta x$

   $i = i + 1$

   }

6. Stop.

   **Problem in true-curve generation algorithm :**

1. To specify a curve, we need more information than just its endpoints. This may mean that a different display-file structure should be used.

2. In tranformations, a line segment when scaled is still a line segment but other curves may behave differently. For example, a circle when scaled in only one direction becomes an ellipse.

3. If we wished to clip arcs to some window boundary, a new clipping algorithm would have to be developed.

**SECTION-C**

3. Attempt any **one** part of the following : **(7 × 1 = 7)**
a. **What are the disadvantages of DDA algorithm ? Also write Bresenham's line drawing algorithm for negative slope.**

**Ans.** **Bresenham's line algorithm :**

1. Input the two lines endpoints and store the left endpoint $(x_0, y_0)$.

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

3. Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$ and $2\Delta y - 2\Delta x$, and obtain the starting value for decision parameter as :

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

4. At each $x_k$ along the line, starting at $k = 0$, perform the following test :

a. If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and

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

b. If $p_k > 0$, the next point to plot is $(x_k + 1, y_k + 1)$ and

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

5. Repeat step 4 $\Delta x$ times.

**Numerical :**

$$\Delta x = x_2 - x_1 = 7 - 1 = 6$$
$$\Delta y = y_2 - y_1 = 6 - 2 = 4$$
$$\text{Incr } c_1 = 2\Delta y = 8$$
$$\text{Incr } c_2 = 2(\Delta y - \Delta x) = 2(4 - 6) = -4$$
$$p_0 = 2\Delta y - \Delta x = 8 - 6 = 2$$
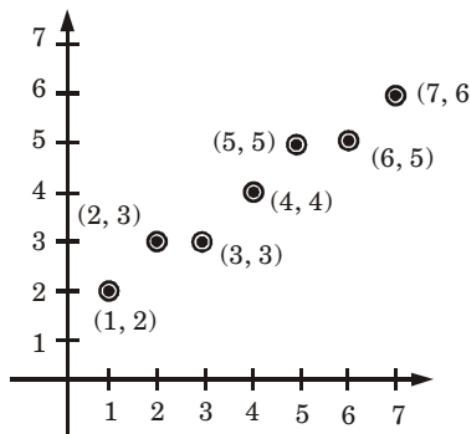
| **Step** | $P_k$ | **Pixel** |
|----------|-------|-----------|
|          | 2     | (2, 3)    |
| 1        | − 2   | (3, 3)    |
| 2        | 6     | (4, 4)    |
| 3        | 2     | (5, 5)    |
| 4        | − 2   | (6, 5)    |
| 5        | 6     | (7, 6)    |

**Fig. 7.**

**Disadvantages of DDA algorithm :**

1. Floating point arithmetic in DDA algorithm is time consuming.
2. Precision loss due to rounding off.
3. Pixel drifts farther apart if line is relatively large.
4. Rounding off in DDA is time consuming.
5. End point accuracy is poor.

   **Bresenham's line drawing algorithm for negative slope :**
   For negative slopes, the procedure is similar, except that one coordinate decreases as the other increases.

**b. Write mid-point circle algorithm and predict the pixels in any octant of circle for radius = 12 pixels with its centre at origin ?**

**Ans.** **Mid-point circle algorithm :**

1. Input radius $r$ and center $(x_c, y_c)$ and obtain the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$ *i.e.*, initialize starting position as

   $$x = 0$$
   $$y = r$$

2. Calculate the initial value of the decision parameter as

   $$p = 1.25 - r$$

3. do
   {
   plot $(x, y)$
   if $(p < 0)$
   {
   $x = x + 1$
   $y = y$
   $p = p + 2x + 1$
   }
   else
   {

$x = x + 1$
$y = y - 1$
$p = p + 2x - 2y + 1$
}
}
while $(x < y)$

4. Determine symmetry points in the other seven octants.
5. Stop.

**Numerical :**

$$r = 12$$

The initial point $(x, y) = (0, 12)$

*i.e.*,  $x = 0, \quad y = 12$

Calculate initial decision parameters $p = 1 - r = 1 - 12 = -11$

$$p = -11 \; (< 0)$$

First plot $(0, 12)$

Here $p < 0$. So, $x = 0 + 1 = 1, \quad y = 12$

and  $p = -11 + 2 \times 1 + 1 = -11 + 3 = -8$

$x < y$ *i.e.*, $1 < 12$. So, condition is true and plot $(1, 12)$.

Now,  $p = -8 \; (< 0)$

so,  $x = 1 + 1 = 2, \quad y = 12$

and  $p = -8 + 2 \times 2 + 1 = -8 + 5 = -3$

$x < y$ *i.e.*, $2 < 12$. So, plot $(2, 12)$.

Now,  $p = -3 \; (< 0)$

so,  $x = 2 + 1 = 3, \quad y = 12$

and  $p = -3 + 2 \times 3 + 1 = 4$

$x < y$ *i.e.*, $3 < 12$. So, plot $(3, 12)$

Now,  $p = 4 \; (> 0)$

so,  $x = x + 1 = 3 + 1 = 4, \quad y = y - 1 = 12 - 1 = 11$

and  $p = p + 2x - 2y + 1 = 4 + 8 - 22 + 1 = -9$

$x < y$ *i.e.*, $4 < 11$. So, the next point to plot is $(4, 11)$.

Now,  $p = -9 \; (< 0)$

so,  $x = x + 1 = 4 + 1 = 5, \quad y = 11$

$p = p + 2x + 1 = -9 + 10 + 1 = 2$

$x < y$ *i.e.*, $5 < 11$. So, the next point plot is $(5, 11)$.

Now,  $p = 2 \; (> 0)$

so,  $x = x + 1 = 5 + 1 = 6, \quad y = y - 1 = 11 - 1 = 10$

$p = p + 2x - 2y + 1 = 2 + 12 - 20 + 1 = -5$

$x < y$ *i.e.*, $6 < 10$. So, plot $(6, 10)$.

Now,  $p = -5 \; (> 0)$

so,  $x = x + 1 = 6 + 1 = 7, \quad y = y - 1 = 10 - 1 = 9$

$p = p + 2x - 2y + 1 = -5 + 14 - 18 + 1 = -8$

We have to continue the iteration until $x^3 y$ *i.e*, either $x > y$ or $x = y$.

**4.** Attempt any **one** part of the following :                    **(7 × 1 = 7)**

**a. Write the Liang Barsky algorithm for line clipping. Use Liang Barsky line clipping algorithm to clip the line $P_1(-1, 7)$ to $P_2(11, 1)$ against the window having diagonally opposite corners as $(1, 2)$ and $(9, 8)$.**

**Ans.** **Liang Barsky algorithm :**

1. Read two endpoint of the line say $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$.

2. Read two corners (left-top and right-bottom) of the window, say $(x_{w\min}, y_{w\max}, x_{w\max}, y_{w\min})$.

3. Calculate the values of parameters $p_i$ and $q_i$ for $i = 1, 2, 3, 4$ such that

   $\quad p_1 = -\Delta x \qquad\qquad\qquad q_1 = x_1 - x_{w\min}$
   $\quad p_2 = \Delta x \qquad\qquad\qquad\ \ q_2 = x_{w\max} - x_1$
   $\quad p_3 = -\Delta y \qquad\qquad\qquad q_3 = y_1 - y_{w\min}$
   $\quad p_4 = \Delta y \qquad\qquad\qquad\ \ q_4 = y_{w\max} - y_1$

4. If $p_i = 0$, then
   {     The line is parallel to $i^{\text{th}}$ boundary.
          if $q_i < 0$ then
       {   line is completely outside the boundary, can be eliminated and goto stop.
       }
       else
       {
       line is inside the boundary and needs further consideration
       }
       else
       {

       Calculate $r_i = \dfrac{q_1}{p_i}$ for $i = 1, 2, 3, 4$

       }
   }

5. For all $k$ such that $p_k < 0$ calculate $r_k = q_k/p_k$. Let $u_1$ be the maximum of the set containing 0 and the various values of $r$.

6. For all $k$ such that $p_k > 0$ calculate $r_k = q_k/p_k$. Let $u_2$ be the maximum of the set containing 1 and the calculated $r$ values.

7. If $u_1 > u_2$, eliminate the line since it is completely outside the clipping window, otherwise use $u_1$ and $u_2$ to calculate the endpoint of the clipped line.

8. Stop.

   **Numerical :**

   $\quad\quad\quad x_1 = -1 \qquad\quad x_{w\min} = 1$
   $\quad\quad\quad y_1 = 7 \qquad\quad\ \ y_{w\min} = 2$
   $\quad\quad\quad x_2 = 11 \qquad\quad x_{w\max} = 9$
   $\quad\quad\quad y_2 = 1 \qquad\quad\ \ y_{w\max} = 8$

   $p_1 = -\Delta_x = -(11 + 1) = -12,\ q_1 = -1 - 1 = -2,\ \dfrac{q_1}{p_1} = \dfrac{-2}{-12} = 0.1667$

$$p_2 = \Delta_x = 11 + 1 = 12, \quad q_2 = 9 + 1 = 10, \quad \frac{q_2}{p_2} = \frac{10}{12} = 0.833$$

$$p_3 = -\Delta_y = -(1 - 7) = 6, \quad q_3 = 7 - 2 = 5, \quad \frac{q_3}{p_3} = \frac{5}{6} = 0.833$$

$$p_4 = \Delta_y = -6, \quad q_4 = 8 - 7 = 1, \quad \frac{q_4}{p_4} = \frac{1}{-6} = -0.1667$$

$t_1 = \max(0.833, 0.1667) = 0.833,$

Since for these values $p < 0$

$$t_2 = \min(0.833, -0.1667) = -0.1667$$

Since for these values $p > 0$

Here, $t_1 < t_2$ and endpoints of clipped line are :

$$xx_1 = x_1 + t_1 \Delta x = -1 + 0.833 \times 12 = 8.996$$
$$yy_1 = y_1 + t_1 \Delta y = 7 + (0.833)(-6) = 2.002$$
$$xx_2 = x_1 + t_2 \Delta x = -1 + (-0.1667) \times 12 = -3.00$$
$$yy_2 = y_1 + t_2 \Delta y = 7 + (-0.1667) \times (-6) = 8.002$$
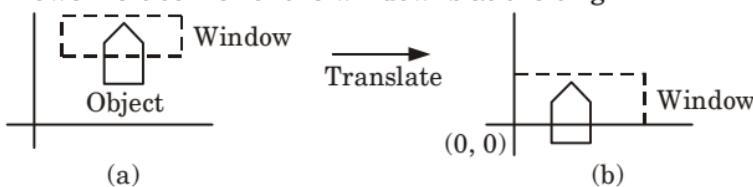
**b. Explain window-to-viewport transformation in detail.**

**Ans.** **Window to view point coordinate transformation :**
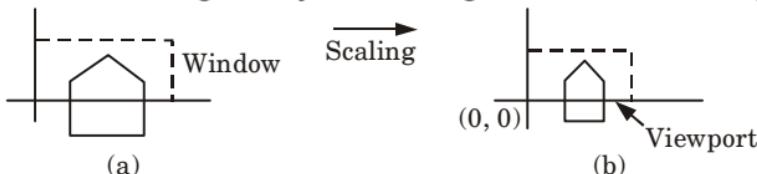
1. Window to viewport / view point Transformation is the process of transforming a 2D world-coordinate objects to device coordinates.
2. Objects inside the world or clipping window are mapped to the viewport which is the area on the screen where world coordinates are mapped to be displayed.
3. Window to viewport mapping or transformation is done in following three steps :
   **Step 1 :** The object together with its window is translated until the lower-left corner of the window is at the origin.



**Fig. 8.**

**Step 2 :** The object and the window are then scaled until the window has the dimension same as viewport. In other words, we are converting the object into image and window in viewport.



**Fig. 9.**

**Step 3 :** The final transformation step is another translation to move the viewport to its correct position on the screen.
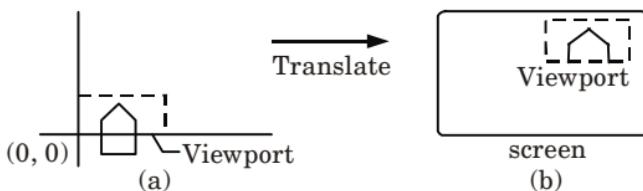
Fig. 10.

**5.** Attempt any **one** part of the following :           **(7 × 1 = 7)**

**a.** A rectangular parallelopiped is given having length on *X*-axis, *Y*-axis and *Z*-axis as 3, 2 and 1 respectively. First apply a rotation of – 90° about the *Y*-axis followed by rotation of 90° about *X*-axis.



Fig. 11.

**Ans.**

$$\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

Rotation matrix about *y*-axis by an angle – 90°
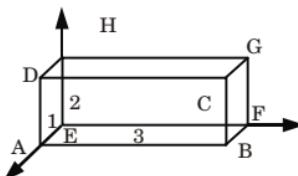
$$\begin{bmatrix} \cos(-90°) & 0 & -\sin(-90°) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(-90°) & 0 & \cos(-90°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, $$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ -1 & 0 & 3 & 1 \\ -1 & 2 & 3 & 1 \\ -1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 2 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix about $x$-axis by angle 90° is given as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90°) & \sin(90°) & 0 \\ 0 & -\sin(90°) & \cos(90°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 3 & 1 & 0 & 1 \\ 3 & 1 & 2 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix}$

So, the new coordinates of rectangular parallelopiped as

(0, 1, 0), (3, 1, 0), (3, 1, 2), (0, 1, 2), (0, 0, 0), (3, 0, 0), (3, 0, 2), (0, 0, 2)

**b. What do you understand by projection ? Differentiate between parallel projection and perspective projection.**

**Ans.** **Projection :** Projection is a process of representing a three dimensional object or scene into two dimensional medium *i.e.*, projection is a shadow of the object.
**Difference :**

| S. No. | Parallel projection | Perspective projection |
|--------|---------------------|------------------------|
| 1. | In parallel projection, parallel lines from each vertex on the object are extended until they intersect the view plane. | Perspective projection produces realistic view. |
| 2. | Lines of projection are parallel. | Lines of projection are not parallel. |
| 3. | These are linear transform. | These are non-linear transform. |
| 4. | In parallel projection, the center of projection is at infinity. | In perspective projections, the center of projection is at a point. |

**6.** Attempt any **one** part of the following :                                  **(7 × 1 = 7)**
**a. Construct the Bezier curve of order 3 and with 4 polygon vertices A(1, 1), B(2, 3), C(4, 3), D(6, 4).**

**Ans.** The equation for the Bezier curve is given as :

$p(u) = (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2 (1-u) P_3 + u^3 P_4$ for $0 \le u \le 1$

where $p(u)$ is the point on the curve $P_1, P_2, P_3, P_4$.

Let us take $\quad u = 0, \dfrac{1}{4}, \dfrac{1}{2}, \dfrac{3}{4}$

$\therefore \qquad P(0) = (1, 1)$

$\therefore \qquad P\left(\dfrac{1}{4}\right) = \left(1 - \dfrac{1}{4}\right)^3 P_1 + 3\dfrac{1}{4}\left(1 - \dfrac{1}{4}\right)^2 P_2$

$$+ 3\left(\dfrac{1}{4}\right)^2 \left(1 - \dfrac{1}{4}\right) P_3 + \left(\dfrac{1}{4}\right)^3 P_4$$

$$= \dfrac{27}{64}(1, 1) + \dfrac{27}{64}(2, 3) + \dfrac{9}{64}(4, 3) + \dfrac{1}{64}(6, 4)$$

$$= \left[\dfrac{27}{64} \times 1 + \dfrac{27}{64} \times 2 + \dfrac{9}{64} \times 4 + \dfrac{1}{64} \times 6, \dfrac{27}{64} \times 1\right.$$

$$\left. + \dfrac{27}{64} \times 3 + \dfrac{9}{64} \times 3 + \dfrac{1}{64} \times 4\right]$$

$$= \left[\dfrac{123}{64}, \dfrac{139}{64}\right] = (1.9218, 2.1718)$$

$\therefore \qquad P\left(\dfrac{1}{2}\right) = \left(1 - \dfrac{1}{2}\right)^3 P_1 + 3\dfrac{1}{2}\left(1 - \dfrac{1}{2}\right)^2 P_2$

$$+ 3\left(\dfrac{1}{2}\right)^2 \left(1 - \dfrac{1}{2}\right) P_3 + \left(\dfrac{1}{2}\right)^3 P_4$$

$$= \dfrac{1}{8}(1, 1) + \dfrac{3}{8}(2, 3) + \dfrac{3}{8}(4, 3) + \dfrac{1}{8}(6, 4)$$

$$= \left[\dfrac{1}{8} \times 1 + \dfrac{3}{8} \times 2 + \dfrac{3}{8} \times 4 + \dfrac{1}{8} \times 6, \dfrac{1}{8} \times 1\right.$$

$$\left. + \dfrac{3}{8} \times 3 + \dfrac{3}{8} \times 3 + \dfrac{1}{8} \times 4\right]$$

$$= \left[\dfrac{25}{8}, \dfrac{23}{8}\right] = (3.125, 2.875)$$

$\therefore \qquad P\left(\dfrac{3}{4}\right) = \left(1 - \dfrac{3}{4}\right)^3 P_1 + 3\dfrac{3}{4}\left(1 - \dfrac{3}{4}\right)^2 P_2$

$$+ 3\left(\dfrac{3}{4}\right)^2 \left(1 - \dfrac{3}{4}\right) P_3 + \left(\dfrac{3}{4}\right)^3 P_4$$

$$= \dfrac{1}{64} P_1 + \dfrac{9}{64} P_2 + \dfrac{27}{64} P_3 + \dfrac{27}{64} P_4$$

$$= \dfrac{1}{64}(1, 1) + \dfrac{9}{64}(2, 3) + \dfrac{27}{64}(4, 3) + \dfrac{27}{64}(6, 4)$$

$$= \left[\dfrac{1}{64} \times 1 + \dfrac{9}{64} \times 2 + \dfrac{27}{64} \times 4 + \dfrac{27}{64} \times 6, \dfrac{1}{64} \times 1\right.$$

$$\left. + \dfrac{9}{64} \times 3 + \dfrac{27}{64} \times 3 + \dfrac{27}{64} \times 4\right]$$

$$= \left[ \frac{289}{64}, \frac{217}{64} \right] = (4.5156, 3.3906)$$

$$P(1) = P_3 = (6, 4)$$

Fig. 12 shows the calculated points of the Bezier curve and curve passing through it.



**Fig. 12.** Plotted Bezier curve.

**b. Write the properties of B-spline curves. Also write advantages of B-spline curves on Bezier curves.**

**Ans.** **Properties of B-splines curves :**

1. The polynomial curve has degree $d-1$ and $C^{d-2}$ continuity over the range of $u$.
2. For $n + 1$ control points, the curve is described with $n + 1$ blending functions.
3. Each blending function $B_{k,d}$ is defined over $d$ subintervals of the total range of $u$, starting at knot value $u_k$.
4. The range of parameter $u$ is divided into $n + d$ subinterval by the $n + d + 1$ values specified in the knot vector.
5. Each section of spline curve is influenced by $d$ control points.
6. Any one control point can affect the shape of atmost $d$ curve sections.
7. For any value of $u$ in the interval from knot value $u_{d-1}$ to $u_{n+1}$ the sum over all basic function is 1 :

$$\sum_{k=0}^{n} B_{k,d}(u) = 1$$

**Advantages of B-spline curves over Bezier curves :**

1. The degree of B-spline polynomial can be set independently of the number of control points.
2. B-spline allows local control over the shape of a spline curve or surface.

**7.** Attempt any **one** part of the following : **(7 × 1 = 7)**

**a. Explain depth buffer method and compare it with A-buffer method.**

**Ans.** **Depth buffer :**
1. $z$-buffer is a simple extension of the frame buffer idea. A frame is used to store the intensity of each pixel in image space.
2. The $z$-buffer is a separate depth buffer used to store the $z$-coordinates or depth or every visible pixel in image space.
3. In this algorithm a buffer of the same size as the frame buffer, is set up which holds depth information.
4. Each elements of the depth buffer corresponds to a pixel in the frame buffer and initially holds the maximum depth in the scene.
5. As each polygon is scan-converted, the depth at each pixel is calculated and compared with the corresponding depth in the depth buffer.
6. If the depth is less than that stored in the depth buffer (*i.e.*, nearer the viewer) then that pixel is set in the frame buffer with the polygon colour at that point and the depth buffer is set to the polygon depth.
7. If the polygon depth is greater (*i.e.*, farther away from the viewer) than the depth buffer at that point is not stored in the frame buffer.
8. The $z$- buffer algorithm can be stated as :
   **Step 1 :** Initialize frame buffer to background colour.
   **Step 2 :** Initialize $z$-buffer to minimum $z$ value.
   **Step 3 :** Scan-convert each polygon in arbitrary order.
   **Step 4 :** For each $(x, y)$ pixel, calculated depth '$z$' at that pixel $(z(x, y))$.
   **Step 5 :** Compare calculated new depth $z(x, y)$ with value previously stored in $z$-buffer at the location $z(x, y)$.
   **Step 6 :** If $z(x, y) > z(x, y)$, then write the new depth value to $z$-buffer and updated frame buffer.
   **Step 7 :** Otherwise, no action is taken.
   **Comparison of depth buffer method with A-buffer method :**
1. The depth buffer (or Z Buffer) method can only be used for opaque object but not for transparent object, the A-buffer method can be used for both objects.
2. A buffer method is slightly costly than depth buffer method because it requires more memory in comparison to the depth buffer method.
3. A buffer method provides anti-aliasing while depth buffer does not.

**b.** **Why is Gouraud shading also referred to as interpolation shading ? Also discuss its advantages and disadvantages.**
**Ans.** **Gouraud shading :**
a. It represents a polygon surface by linearly interpolating intensity across the polygon surface.
b. Intensity values for each polygon are matched with adjacent polygon along common edges.
c. It eliminates the intensity discontinuity that can occur in constant-intensity shading.

    d. Gouraud shading is a trick whereby values for lighting, fog, and so forth are computed at the vertex level. These values are then linearly interpolated across the face of the polygon.

    e. Due to the use of linear interpolation Gouraud shading is also referred to as interpolation shading.

**Advantage :**

   i. Gouraud shading can be combined with a hidden surface algorithm to fill in the visible polygons along each scan line.

  ii. It removes the intensity discontinuities exists in constant shading model.

**Disadvantages :**

   i. Highlights on the surface are sometimes displayed with anomalous shapes.

  ii. The mach band effect can be reduced by breaking the surface into a greater number of smaller polygons.

 iii. Sharp drop of intensity values on the polygon surface cannot be displayed.

☺☺☺