1 UNIT

Introduction and Line Generation

CONTENTS

Part-1	:	Introduction and Line Generation : Types of Computer Graphics	1-2C – 1-3C
Part-2	:	Graphics Displays-Random Scan Displays Raster Scan Displays	1-3C – 1-10C
Part-3	:	Frame Buffer and Video Controller	1-10C – 1-14C
Part-4	:	Points and Lines	1-14C – 1-17C
Part-5	:	Line Drawing Algorithm	1-17C – 1-25C
Part-6	:	Circle Generating Algorithm Mid-point Circle Generating Algorithm & Parallel Version of these Algorithm	1-25C – 1-32C

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.

Scanned by CamScanner

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.

Scanned by CamScanner

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

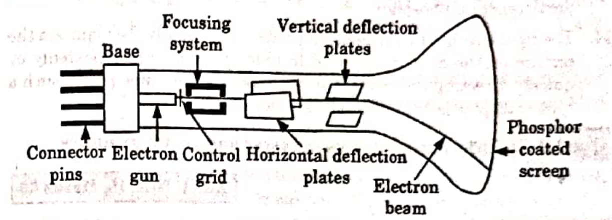


Fig. 1.7.1. CRT.

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.

Scanned by CamScanner

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.

Scanned by CamScanner

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.

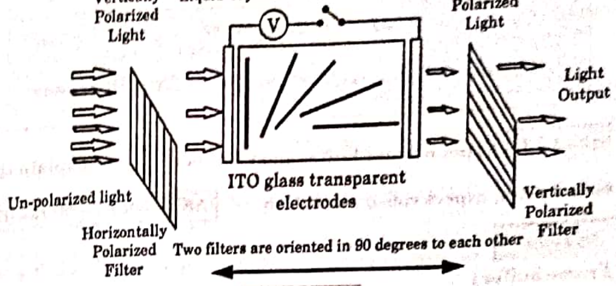


Fig. 1.11.1.

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.

Scanned by CamScanner

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 :

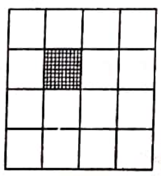
1. Frame buffer is a special area of memory in raster displays which is dedicated to graphics.
2. It holds the set of intensity values for all the screen points.
3. The stored intensity values are retrieved from frame buffer and displayed on the screen one row (scanline) at the time.
4. Each screen point 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.

Scanned by CamScanner

Therefore, time taken by 1024 rows = 1/60 sec.
Hence, time taken by each row of pixels to refresh
$= 1/(60 \times 1024) = 0.0162 \times 10^{-3}$ 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 = 460800 bytes	= (640 × 480 × 24)/8 = 921600 bytes
1024 × 1280	= (1024 × 1280 × 12)/8 = 1966080 bytes	= (1024 × 1280 × 24)/8 = 3932160 bytes
2560 × 2048	= (2560 × 2048 × 12)/8 = 7864320 bytes	= (2560 × 2048 × 24)/8 = 15728640 bytes

Que 1.17. **Consider two raster systems with resolutions of 640 x 480 and 1280 x 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

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

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

say 1 dot = n bits

Transfer rate = 1 mips

= 1×10^6 word per second (1 mips = 10^6 word)

= 32×10^6 bits/second (1 word = 32 bits)

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

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

Scanned by CamScanner

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

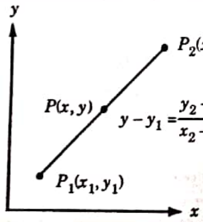


Fig. 1.20.1.

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

where $\qquad m = \frac{y_2-y_1}{x_2-x_1}$

$$y - y_1 = mx - mx_1$$

$$y = mx - mx_1 + y_1$$

$$y = mx + (-mx_1 + y_1)$$

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

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

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

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

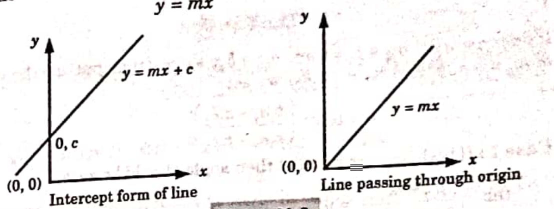


Fig. 1.20.2.

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

$$Ax + By + C = 0$$

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 $\theta_2 - \theta_1$)

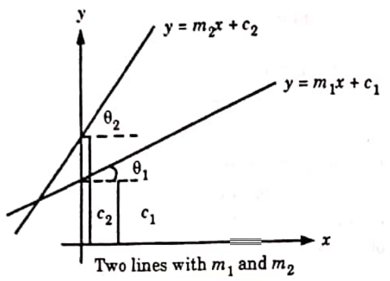
Taking tan both sides

$$\tan\theta = \tan(\theta_1 - \theta_2)$$

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

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

Scanned by CamScanner



Two lines with m_1 and m_2

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.*, slopes are equal.

Similarly, $a_1b_2 = b_1a_2$

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

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

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

$$1 + m_1.m_2 = 0$$

$$m_1.m_2 = -1$$

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

$$a_1a_2 + b_1b_2 = 0$$

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 Δx or Δy.
- 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
   length = Δx
   else
   length = Δy
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.

Scanned by CamScanner

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 ($i \le$ length)

{

Plot (integer (x), integer (y))

$x = x + \Delta x$

$y = y + \Delta x$

$i = i + 1$

}

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)

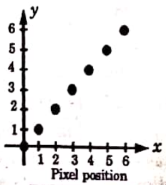


Fig. 1.22.1.

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

AKTU 2014-15, Marks 06

Answer

$$(x_1, y_1) = (-3, 0) \quad , \quad (x_2, y_2) = (4, 4)$$

$$\Delta x = 4 - (-3) = 7$$

$$\Delta y = 4 - 0 = 4$$

$$\text{Length} = \Delta x \qquad (\because \Delta x > \Delta y)$$

$$\Delta x = \frac{4-(-3)}{7} = 1$$

$$\Delta y = \frac{4-0}{7} = \frac{4}{7} = 0.57$$

$$x = x_1 + 0.5 \times \text{sign}(\Delta x) = -3 + 0.5 \times 1 = -2.5$$

$$y = y_1 + 0.5 \times \text{sign}(\Delta y)$$

$$= 0 + 0.5 \times 0.57 = 0.28$$

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)

$$x = -0.5 + 1 = 0.5$$

$$y = 1.42 + 0.57 = 1.99$$

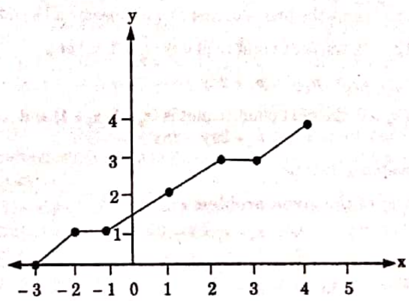


Fig. 1.23.1.

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 Δx, Δy, $2\Delta y$ 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 Δx 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)

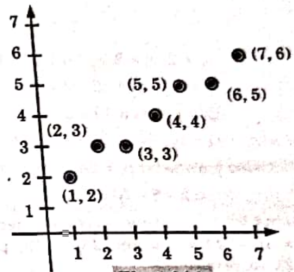


Fig. 1.24.1.

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 d[illegible] not round off but ta[illegible] incremental value operation.

Scanned by CamScanner

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)

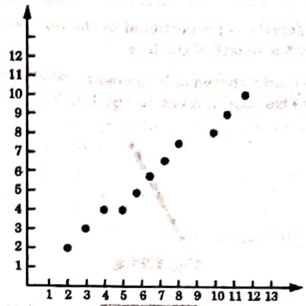


Fig. 1.26.1.

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.

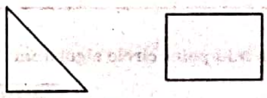


Fig. 1.27.1.

3. **Line should have constant density :**
   a. To maintain the constant density throughout the line, dots should be equally spaced.

Scanned by CamScanner

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

Scanned by CamScanner

Answer

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

Numerical :

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 $p = -10(< 0)$

So $x = 1 + 1 = 2$

$y = 14$

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

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

Now $p = -5(< 0)$

So $x = 2 + 1 = 3$

$y = 14$

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

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

Now $p = 2(> 0)$

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

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

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

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

$= 2 + 8 - 26 + 1$

$= 10 - 26 + 1 = -15$

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

Now $p = -15(< 0)$

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

$y = 13$

$p = p + 2x + 1$

$= -15 + 10 + 1 = -4$

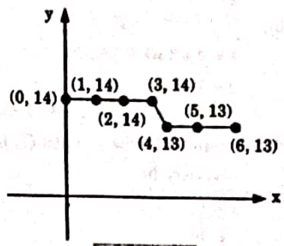


Fig. 1.29.1.

$x < y$, 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 $(x, y) = (0, 10)$

Scanned by CamScanner

$x = 0$

$y = 10$

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

$p = -9 (< 0)$

First plot (0, 10) $\quad x = 0 + 1 = 1$

Here $p < 0$ so $\quad y = 10$

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

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

$p = -6 (< 0)$

Now $\quad x = 1 + 1 = 2$

So $\quad y = 10$

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

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

$p = -1 (< 0)$

Now $\quad x = 2 + 1 = 3$

So $\quad y = 10$

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

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

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

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

$y = y - 1 = 10 - 1 = 9$

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

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

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

So $\quad x = x + 1 = 4 + 1 = 5$

$y = y = 9$

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

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

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

So $\quad x = x + 1 = 5 + 1 = 6$

$y = y - 1 = 9 - 1 = 8$

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

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

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

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

$y = y - 1 = 8 - 1 = 7$

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

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

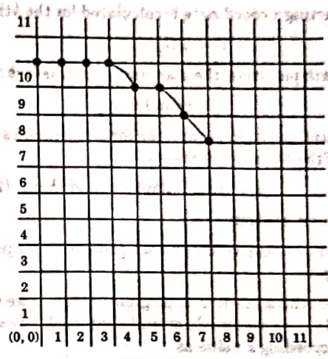


Fig. 1.30.1.

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.

Scanned by CamScanner

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 Δx is the width of the line and the value for partition width Δx_p is computed using integer division.

d. The starting x coordinate is calculated for the kth partition as :

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

e. At the kth 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 kth 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, $$A = \frac{-\Delta y}{\text{linelength}}$$

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

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

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

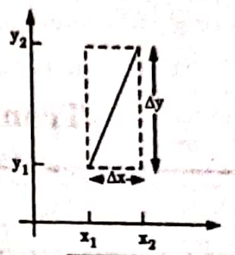


Fig. 1.31.1.

☺☺☺

2 UNIT

Transformation

CONTENTS

Part-1 :	Transformations : Basic Transformation Matrix Representation and Homogeneous Coordinates	2-2C - 2-5C
Part-2 :	Composite Transformation	2-6C - 2-9C
Part-3 :	Reflections and Shearing	2-9C - 2-18C
Part-4 :	Windowing and Clipping : Viewing Pipeline Viewing Transformation	2-18C - 2-20C
Part-5 :	2-D Clipping Algorithm-Line Clipping Algorithm such as Cohen Sutherland Line Clipping Algorithm Liang-Barsky Algorithm	2-20C - 2-30C
Part-6 :	Line Clipping against Non-Rectangular Clip Window Polygon Clipping-Sutherland Hodgeman Polygon Clipping Weiler and Atherton Polygon Clipping	2-30C - 2-37C
Part-7 :	Curve Clipping Text Clipping	2-37C - 2-39C

PART-1

Transformations : Basic Transformation.

CONCEPT OUTLINE

- 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.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

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.

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

   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 :

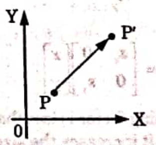


Fig. 2.1.1.

$$P = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

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

$$T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Hence $\quad P' = P + T$

2. Scaling :

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

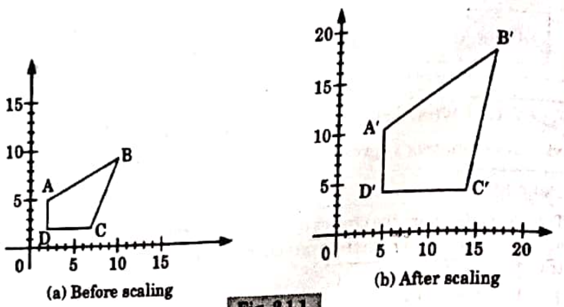


Fig. 2.1.1.

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.

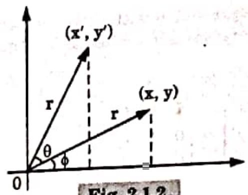


Fig. 2.1.2.

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 :

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

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

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

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

Scanned by CamScanner

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

Que 2.2. What do you understand by homogeneous coordinates ?

Answer

1. Homogeneous coordinates are defined as the coordinates that represent all the geometric transformation equations as matrix multiplication.
2. Coordinates are represented with three element column vectors, and transformation operations are written as 3 × 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 θ 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 :

Scanned by CamScanner

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

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

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

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

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

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 :

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

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

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

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

Scanned by CamScanner

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

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 θ 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 θ 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 $(0, b)$.

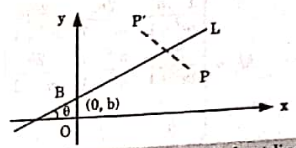


Fig. 2.6.1. Reflection of point P about line L.

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

Scanned by CamScanner

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

After multiplication of all basic matrix, we get

$$M_L = \begin{bmatrix} \frac{1-m^2}{m^2+1} & \frac{2m}{m^2+1} & \frac{-2bm}{m^2+1} \\ \frac{2m}{m^2+1} & \frac{m^2-1}{m^2+1} & \frac{2b}{m^2+1} \\ 0 & 0 & 1 \end{bmatrix}$$

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 $T_R \rightarrow$ Translation matrix

$R_\theta \rightarrow$ Rotation matrix by an angle $\theta°$

$[T_R]^{-1} \rightarrow$ 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

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

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 :

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

Scanned by CamScanner

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

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

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

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

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

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

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

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

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

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

ii. Vertical line $X = 2$:

$$X = 2$$

$$y = mx + c$$

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

$$m = \infty, c = -2$$

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

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

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

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

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

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

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

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

Scanned by CamScanner

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

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

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.**

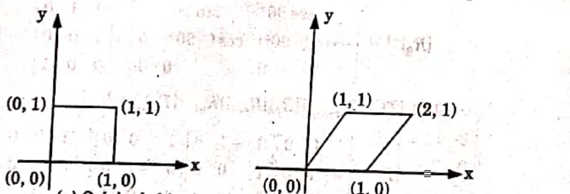
Answer

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 :

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

$$x' = x + Sh_x.y$$
$$y' = y$$

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



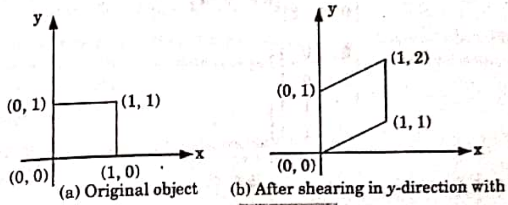
(a) Original object (b) After shearing in x-direction with $Sh_x = 1$

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 :

$$Sh_{yref} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & -Sh_y.x_{ref} \\ 0 & 0 & 1 \end{bmatrix}$$

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

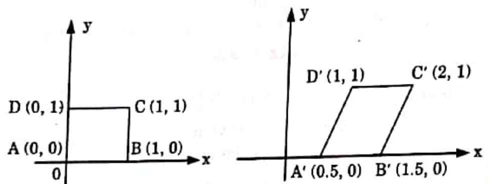
Scanned by CamScanner

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 $y_{ref} = -1$

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

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

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



(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

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

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 =

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

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.

Scanned by CamScanner

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.

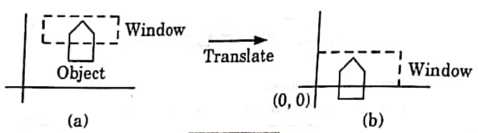


Fig. 2.13.1.

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.

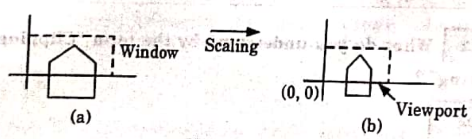


Fig. 2.13.2.

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

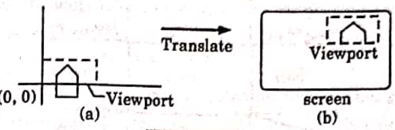


Fig. 2.13.3.

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.

Scanned by CamScanner

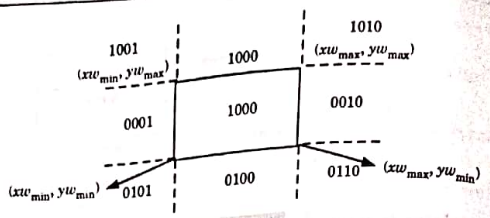


Fig. 2.14.1.

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 ($X_{wmin} = Y_{wmin} = 5$ and $X_{wmax} = Y_{wmax} = 10$).** **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.

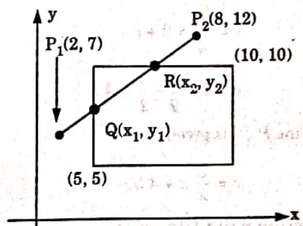


Fig. 2.15.1.

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 : $x - x_{wmin}$	$x - 5$	-3	3
Bit 2 : $x_{wmax} - x$	$10 - x$	8	2
Bit 3 : $y - y_{wmin}$	$Y - 5$	2	7
Bit 4 : $y_{wmax} - y$	$10 - y$	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

Scanned by CamScanner

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 $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{12-7}{8-2} = \frac{5}{4}$

Equation of line P_1P_2 is given as

$$y = mx + c$$

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

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

$\because$ $P(2, 7)$ 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

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

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

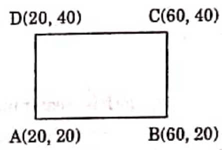


Fig. 2.16.1.

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.

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

Scanned by CamScanner

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.**

($x_{wmin} = y_{wmin} = 5$ and $x_{wmax} = y_{wmax} = 10$). **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 $(x_{wmin}, y_{wmax}, x_{wmax}, y_{wmin})$.
3. Calculate the values of parameters p_i and q_i for $i = 1, 2, 3, 4$ such that

$p_1 = -\Delta x$ $\quad q_1 = x_1 - x_{wmin}$

$p_2 = \Delta x$ $\quad q_2 = x_{wmax} - x_1$

$p_3 = -\Delta y$ $\quad q_3 = y_1 - y_{wmin}$

$p_4 = \Delta y$ $\quad q_4 = y_{wmax} - y_1$

4. If $p_i = 0$, then
```
{   The line is parallel to ith 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 = q_i / 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 :

$$P_1 = (2, 7)$$
$$P_2 = (8, 12)$$
$$(x_{wmin}, y_{wmin}) = (5, 5)$$
$$(x_{wmin}, y_{wmax}) = (5, 10)$$

Scanned by CamScanner

$$(x_{wmax}, y_{wmin}) = (10, 5)$$
$$(x_{wmax}, y_{wmin}) = (10, 10)$$
$$\Delta x = 8 - 2 = 6$$
$$\Delta y = 12 - 7 = 5$$

$p_1 = -\Delta x = -6$	$q_1 = x_1 - x_{wmin} = 2 - 5 = -3$
$p_2 = \Delta x = 6$	$q_2 = x_{wmax} - x_1 = 10 - 2 = 8$
$p_3 = -\Delta y = -5$	$q_3 = y_1 - y_{wmin} = 7 - 5 = 2$
$p_4 = \Delta y = 5$	$q_4 = y_{wmax} - y_1 = 10 - 7 = 3$

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

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

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

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

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

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

$$u_1 < u_2$$

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

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

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

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

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

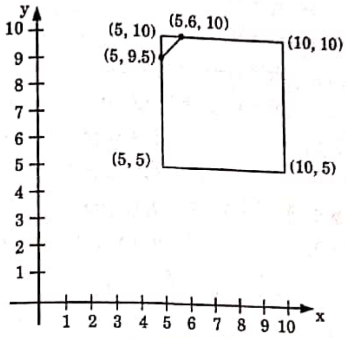


Fig. 2.17.1.

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

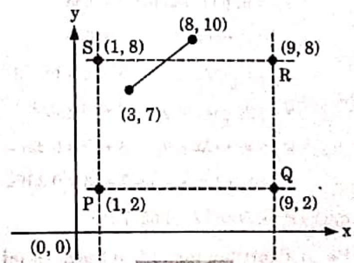


Fig. 2.18.1.

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

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

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

Scanned by CamScanner

$(x_1, y_1) = (3, 7)$

$(x_2, y_2) = (8, 10)$

$\Delta x = x_2 - x_1 = 8 - 3 = 5$

$\Delta y = y_2 - y_1 = 10 - 7 = 3$

$p_1 = -\Delta x = -5 \qquad q_1 = x_1 - x_{min} = 3 - 1 = 2$

$p_2 = \Delta x = 5 \qquad q_2 = x_{max} - x_1 = 9 - 3 = 6$

$p_3 = -\Delta y = -3 \qquad q_3 = y_1 - y_{min} = 7 - 2 = 5$

$p_4 = \Delta y = 3 \qquad q_4 = y_{max} - y_1 = 8 - 7 = 1$

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

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

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

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

$u_1 = \max(-0.4, -1.66, 0) = 0$

$u_2 = \min(1, 1.2, 0.33) = 0.33$

$u_1 < u_2$ endpoints are visible

$x'_1 = x_1 + (\Delta x \times u_1) = 3 + (5 \times 0) = 3$

$y'_1 = y_1 + (\Delta y \times u_1) = 7 + (3 \times 0) = 7$

$x'_2 = x_1 + (\Delta x \times u_2) = 3 + (5 \times 0.33) = 4.65$

$y'_2 = y_1 + (\Delta y \times u_2) = 7 + (3 \times 0.33) = 7.99$

$\therefore$ 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.**

Scanned by CamScanner

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.

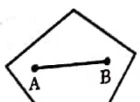


Fig. 2.20.1.

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

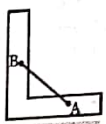


Fig. 1.20.2.

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.

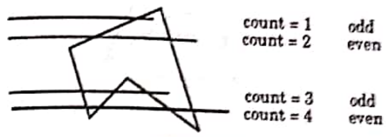


Fig. 2.21.1.

      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.

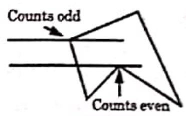


Fig. 2.21.2.

   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.

Scanned by CamScanner

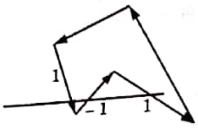


Fig. 2.21.3.

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.

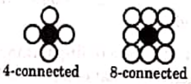


Fig. 2.22.1.

```
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.**

Scanned by CamScanner

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.

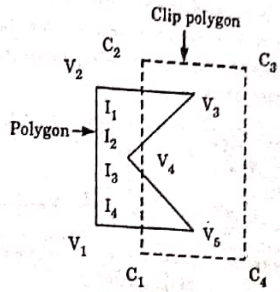


Fig. 2.23.1.

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 :

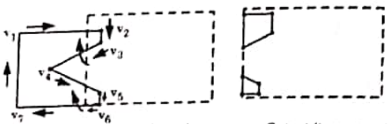
1. Read coordinates of all vertices of the polygon.
2. Read coordinates of the clipping window.
3. Consider the left edge of the window.
4. Compare the vertices of each edge of the polygon, individually with the clipping plane.
5. Save the resulting intersections and vertices in the new list of vertices 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.

Scanned by CamScanner

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 applying Weiler-Atherton algorithm

Output (two separate 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.

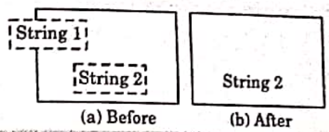
vi. If the two regions overlap, we will need to solve the simultaneous line-curve equations to obtain the clipping intersection points.

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.



(a) Before (b) After

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.

Scanned by CamScanner

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

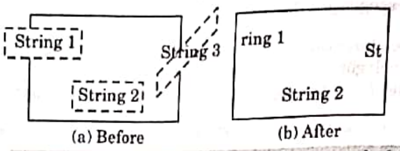


Fig. 2.25.2. All-or-none character clipping method.

☺☺☺

3 UNIT
Three Dimensional

CONTENTS

Part-1 :	Three Dimensional : 3D Geometric Primitive, 3D Object Representation	3–2C – 3–3C
Part-2 :	3D Transformation, 3D Viewing	3–3C – 3–8C
Part-3 :	Projection	3–8C – 3–17C
Part-4 :	3D Clipping	3–18C – 3–23C

Scanned by CamScanner