PGScIT (S4) 02
Exam. Code: M4CGR

Computer Graphics

SEMESTER-IV

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

BLOCK-1

KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Shikhar Kumar Sarma, Gauhati University

Dr. Jyoti Prokash Goswami, Assam Engineering College

Programme Coordinator

Dr. Ridip Dev Choudhury, KKHSOU

Course Coordinators

Dr. Tapashi Kashyap Das, KKHSOU

Dr. Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
1, 2. 3. 4. 5	Mr. Dipankar Dutta, NERIM, Guwahati
6, 7	Mrs. Pinki Saikia Dutta, GIMT, Guwahati

Editorial Team

Content : Dr. Jonalee Barman Kakati, NERIM, Guwahati

Structue, Format & Graphics : Dr Nabankur Pathak, KKHSOU

Aug, 2022
ISBN NO :

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international): http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Headquarters : Patgaon, Rani Gate, Guwahati - 781017
City Office : Resham Nagar, Khanapara, Guwahati- 781022
Web: www.kkhsou.in

COURSE INTRODUCTION

This is a course on ***Computer Graphics***. One of the most natural ways to interact with a computer is through graphics. The use of a computer to define, store, manipulate, and show visual output is known as computer graphics. It includes the production and display of basic graphic elements and images as well as contemporary methods for creating a virtual reality. One needs to have a rudimentary understanding of the fundamental ideas behind computer graphics in order to properly utilise these techniques. There are several applications for computer graphics in business, government, education, entertainment, and most recently, the home.

This course covers the various basic graphics concepts and techniques for operating with two and three-dimensional objects and overview a few typical applications of computer graphics. The course is divided into two blocks

Block 1 deals with the basic concepts of computer graphics and graphics system. Along with this, it also elaborates about various algorithm like Line drawing, Circle and Ellipse drawing, field area. It also explains about transformations and viewing

Block 2 deals with clipping algorithms, 3 D object representation, Bezier and B-spline curves, Projections, Geometric transformation, 3-D viewing and basic animations

BLOCK INTRODUCTION

This is the first block of the course **Computer Graphics**. This block describes the basic principles, techniques, algorithms for generating and interacting with simple graphical objects on a display screen. Two-dimentional transformations and two-dimentional viewing are described in this block.

This block comprises the following seven units:

Unit 1 is an introductory unit which provides the definition and application areas of Computer Graphics. It also gives the concept of Graphical User Interface.

Unit 2 describes various graphics systems such as Cathod Ray Tube, Raster Scan and Random Scan Display system and Graphics Software.

Unit 3 presents important line drawing algorithms. It mainly focuses on DDA and Bresenham's line drawing algorithm.

Unit 4 presents Circle and Ellipse drawing algorithms.

Unit 5 discusses Filling algorithms. Boundary-fill, Flood-fill algorithms are described in this unit.

Unit 6 focuses on two-dimensional transformations. Homogeneous coordinates, composition of 2D transformations are described in this unit.

Unit 7 is on two-dimensional viewing. The viewing pipeline, window-to-viewport coordinate transformation in matrix form, viewing functions are described in this unit.

Each unit of this block includes some along-side boxes to help you know some of the difficult, unseen terms. Some "EXERCISES" have been included to help you apply your own thoughts. You may find some boxes marked with "LET US KNOW". These boxes will provide you with some additional interesting and relevant information. Again, you will get "CHECK YOUR PROGRESS" questions. These have been designed for you to self-check your progress of study. It will be helpful for you if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with "ANSWERS TO CHECK YOUR PROGRESS" given at the end of each unit.

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

Computer Graphics

DETAILED SYLLABUS

Page No.

Unit 1: Introduction to Computer Graphics [5 marks] **7-24**

Definition, Application Areas of Computer Graphics (CAD, CAM, Education and Training, Entertainment, Image Processing, Computer Art etc.), Graphical User Interfaces (GUI)

Unit 2: Graphics Systems [5 marks] **25-47**

Cathode Ray Tubes, Random Scan Displays, Raster Scan Displays, Color CRT Monitors, FlatPanel Displays (Plasma-Panels, Liquid Crystal Displays(LCD), Electroluminescent displays), Graphics Software (GKS, PHIGS)

Unit 3: Line Drawing Algorithms [7 marks] **48-65**

Points and Lines, Line Drawing Algorithms (DDA Algorithm, Bresenham s Line Algorithm)

Unit 4: Circle and Ellipse Drawing Algorithms [7 marks] **66-89**

Circle drawing algorithms, Ellipse Drawing algorithms

Unit 5: Filled area algorithms [6 marks] **90-110**

Filling (Scan-Line Polygon filling, Inside outside tests, Boundary-fill and Flood-fill algorithm)

Unit 6: Transformations [8 marks] **111-121**

Basic 2-D Transformations (Rotation, Reflection, shearing, scaling), Homogeneous Coordinate Representation, Translation, 3-D transformations

Unit 7: 2-D Viewing [8 marks] **122-130**

2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing function

UNIT 1: INTRODUCTION TO COMPUTER GRAPHICS

UNIT STRUCTURE

1.1 Learning Objectives

1.2 Introduction

1.3 Definition of Computer Graphics

1.4 History of Computer Graphics

1.5 Applications of Computer Graphics

- 1.5.1 Computer-aided design
- 1.5.2 Presentation graphics
- 1.5.3 Computer Art
- 1.5.4 Entertainment
- 1.5.5 Education and training
- 1.5.6 Visualization
- 1.5.7 Image Processing
- 1.5.8 Graphical User Interface

1.6 Input Devices

1.7 Output Devices

1.8 Let Us Sum Up

1.9 Further Reading

1.10 Answer to Check Your Progress

1.11 Model Questions

1.1 LEARNING OBJECTIVES

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

- Describe about the computer graphics.
- Classify the application of computer graphics
- Explain about graphical user interface.
- Identify Input/ output devices used in computer graphics.

1.2 INTRODUCTION

This is the first Unit of this block. In this unit we will discus about the concept computer graphics. In the era of digital science, computer graphics becomes an efficient tool for producing pictures, motion pictures, animations, and multimedia. Nowadays, in each field, the use of computer graphics has been seen and there is no area which is left untouched by the use of graphics. Computer graphics are used in diverse areas such as science, medical, engineering, business, medicine, art, government, industry, entertainment, advertising, education and training, simulation, graph representation, printing, etc. Simulation of different experimental environments, training environments, and adventures environment are the most useful and popular use of computer graphics. An electronic circuit can be designed and developed using a simulator and can be tested in the simulation environment. After achieving success, one can move on to experiment with the real equipments. Similarly, experiments in a chemical laboratory can be implemented through simulation and it saves lots of raw materials and equipments from being damaged. Smart classes are now the most popular mode of teaching at the school level. The use of Information and Communication Technology (ICT) are increasing at all level of teacher education and the learning process. Simulation of car driving, airplane pilot training, automation, GPS tracking system, e-business, biology, different print media, and lots more are the broad application of computer graphics. There are hardly a few elements of our lives today that are not influenced by computer graphics. In the next unit we will discuss about graphic systems

1.3 DEFINITION OF COMPUTER GRAPHICS

Computer graphics is primarily concerned with image generation, with applications ranging from entertainment to scientific visualization.Computer graphics can be defined as a rendering tool for the creation and manipulation of images. Thus, computer graphics is defined

as the use of computer as a rendering tool for the development and manipulation of images. Computer graphics is the art of drawing pictures or images with the help of programming.It involves data processing, creation, and computation. Any sketch, drawing, unique artwork, or other material created with the use of a computer to pictorially describe an object or a process or transmit the information as a complement to or replacement of written descriptions is referred to as computer graphics. The base of each drawing made by computer graphics is some mathematical expression or a mathematical model. Commonly the computer graphics system represents 3-dimensional objects in a 2-dimensional viewport. To view or to project a 3-dimensional object into a 2-dimensional viewing window, lots of mathematical computation steps are required including coordinate geometry, trigonometry, and the concepts of the Bayesian Curve.

1.4 HISTORY OF COMPUTER GRAPHICS

In the year 1950, BenLaposky used an electronic machine to create an image of an Oscilloscope. It was considered the first graphics made by a human using an electronic machine they attempted to display two real-time video streams with the whirlwind computer at MIT in 1951. The term "Computer graphics" was coined by William Fetter to produce new design techniques in the year 1960. If anyone wanted to make computer graphics back in the day, they had to design and write program statements that dealt directly with the graphics hardware. Many "graphics libraries" were established to remedy this problem, but most of them were designed for specific operating systems and could not be easily converted to different operating systems or hardware. In 1961, the first video game "Space War" was launched by Steve Russel. Evan Sutherland built the first computer graphics design system as part of his Ph.D. thesis at MIT in 1963. Jack Bresenham proposed a line drawing algorithm in the year 1965. In the year 1969, The Bell Lab designed a pixel with three bits for the first time. In the early days of 1970, raster scan display devices came into the picture. PHIGS was an attempt by the ISO (International Organization for Standardization)

to produce a standard computer graphics library in the 1980s. 1974, Edwin Catmuff developed the texture mapping and the Z-Buffer Hidden-Surface algorithm. Since the development of VGA and SVGA in the 1990s, personal computers could show photo-realistic images and videos with ease. Silicon Graphics was the leading computer graphics firm in the late 1980s and early 1990s. As an alternative to PHIGS, Silicon Graphics released their graphics library (GL) as an "open" standard named OpenGL in 1992. OpenGL is still the only real-time 3D graphics standard that works on a variety of Operating Systems. The computer graphics devices required a 3D graphics solution, andas a result, OpenGL ES (Embedded Subset) was created by Silicon Graphics in the year 1993 and it evolves continuously from version 1.0 to version 3.2."WebGL (Web Graphics Library)" is a JavaScript API that allows any compatible web browser to render interactive 3D computer graphics and 2D graphics without the use of plug-ins.WebGL is integrated into all of the browsers that follow web standards, allowing GPU-accelerated physics, image processing, and effects to be used as part of the web page canvas. In 1993, Steven Spielberg made the fiction film "Jurassic Park" using computer graphics.

1.5 APPLICATION OF COMPUTER GRAPHICS

In computer graphics, an image or a picture consists of pixels. Pixel is the tiniest element of a picture. All the information regarding a pixel is stored in a frame buffer using 24-bit. It is said that a pixel is worth a thousand words. Nowadays graphics are necessary everywhere to help people understand things, thoughts, and ideas.Almost every aspect of our lives benefits from computer graphics. This section covers some of the most prominent computer graphics applications.

1.5.1 Computer-Aided Design

Mostly computer graphics are used in the design process such as engineering, and architectural design. Buildings, airplanes, spacecraft, textiles, watercraft, automobiles, and many other areas

benefit from computer-aided design. The software package CAD is typically used to design images with the features of a multi-window environment. Circuits related to network architecture, water supply, or any other utilities can be drawn with the repeated placement of some common graphical shapes. Standard shapes for communication networks, electrical circuits, water supply, and logic



Figure 1: Architectural design using CAD package (Source: Internet)

often provided by graphics packages. Instead of these, personalized or customized symbols are also useful in these packages. The graphics systems provide an automatic connection between the designed elements through which the designers can easily generate the required output picture. In a CAD application, animation can be used to get the feel of real-time wireframe displays on a video monitor. Wireframe displays allow a designer to view the interior designof an object and also the interior and exterior motions made by the object or the object's components. It is also possible to observe the behaviour of different inner components of a vehicle



Figure 1: Automobile and Mechanical Engineering design using CAD package (Source: Internet)

Figure 1 and 2 depict some of the application of computer graphics in the architectural design of a car and machinery components used

in mechanical engineering. After completion of the object design, a realistic view of the object can be obtained by applying surface rendering. Surface rendering displays the overall appearance of the final project. Using unique light effects and backgrounds, these realistic displays are utilized in product presentations and advertising automobiles and other vehicles.Graphical design can be utilized in the production of mechanical parts by defining how the surfaces of an object will be created with machine tools during fabrication.Some numerically enhanced tools can be used to create the machine parts according to this layout. Building architects used the graphics tools to design the blueprint of a building, flour-plane, positions of doors, windows, passage, electric equipment, furniture, different safety measures, etc. before construction. The birds-eye



Figure 1: Depicts two architectures of buildings using CAD (Source: Internet)

view of the building can be designed to look at the overall structure or the appearance of the building from the top that is from the sky. Electrical and other wiring design also can be built through computer graphics. Figure 3 shows some architectural designs of buildings drawn using graphics applications.

1.5.2 Presentation Graphics

Presentation is one of the most important parts of our daily work. To present a report on any survey, financial report, scientific, economic mathematical, statistical, research data, managerial report, consumer information bulletin, health, transportation, population, educational, the launch of a new product, new design

of a car, building, etc. we can use computer graphics. Nowadays, the most common application of computer graphics is presentation. During teaching, the use of a projector to present study materials in front of the students is a common tool. The presentation may contain text files in different formats, images, videos, audio files, animation, etc. The summary of a research or survey can be presented through some mathematical data, pie-chart, bar diagrams, line graphs, and many more ways.

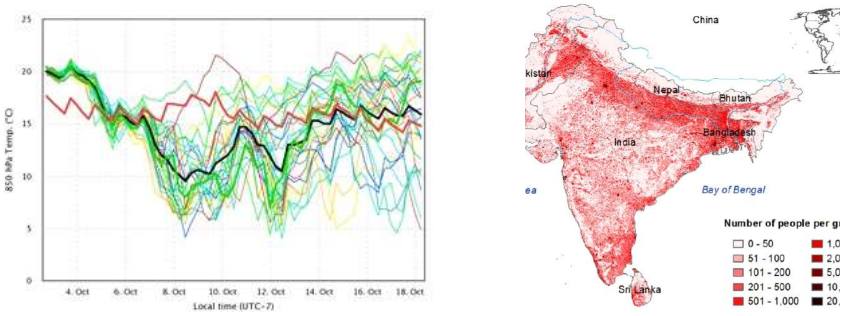


Figure 1: Depicts a weather report and geographical population density of India (Source: Internet)

1.5.3 Computer Art

Computer graphics are widely used in fine art and commercial art applications. Artists use different graphical tools to create their digital canvas such as special purpose hardware, Lumena, PixelPaint, SuperPaint, multimedia packages, CAD packages, animation packages, etc. With the help of such applications, artists can specify shapes and colours for their imaginations, and also specify motion on it.

Other computer technologies are used by fine art artists to create their images.3-D



Figure 1: Drawing of some pictures using graphics tools (Source: Internet)

modeling software, texture mapping, drawing applications, and CAD software are just a few of the technologies used by fine art artists. Modern graphics tools can use to generate electronic designs such as commercial art for logos, page layouts, advertisements, etc. Animation is frequently used in advertising or the creation of cartoon series for kids. "Tom and Jerry", "ChotaBhim", and "Doremon" are some popular animated cartoon series telecast on television. Morphing is another common application of computer graphics, where one object is transformed into another.

1.5.4 Entertainment

Computer graphics are commonly employed in the production of films, music videos, and television shows.Some scenes are fully made out of visuals of graphics only, while others combine actors with graphical scenes. "Tom and Jerry" is a good example of a motion picture (Popular Television Show) that is entirely used the graphics method. "Jurassic Park", "Spiderman", "Harry Potter", etc. are examples of motion pictures where graphics scenes are made with actors. Similarly, graphics scenes are used in music videos also.



Figure 1: Images from animated motion pictures (Source: Internet)

1.5.5 Education And Training

In the 21st century, education without computer graphics is almost nill. Students can learn most of the things or concepts through animation or graphical representation of any scientific phenomenon, economic analysis and biological events over the internet. The concept of smart classes is widely used in school-level learning. Through smartclasses, students can perform their laboratory

activities for physics, chemistry, biology, etc., and experience different scientific phenomena virtually. That is without going to a physical lab, they can learn from this virtual platform.



Figure 7: Simulation of a Pilot training (Source Internet)



Figure 8: Simulation of Ship-Captain Training (Source: Internet)



Figure 9: Simulation of automobile opeator Training (Source: Internet)



Figure 10: Simulation of Solar system (Source: Internet)

This method helps the students a lot because for all students of a class maybe, all the laboratory experiments are not possible to do physically or it is not possible to experience all the scientific phenomena in a classroom environment. For example how the earth revolves around the Sun or how the other planets are revolving around the Sun, how a nuclear reactor works, etc. can be present through animation. The visual scenes make the classroom environment very easy and comfortable for students. Graphical tool simulators are successfully used in different training such as training of ship captain, training for a pilot, air-traffic control, heavy equipment operators, automobile driving, etc. In a normal classroom teaching, teachers used a projector to present their slides containing different graphics or multimedia. Visualization makes learning very comfortable and fruitful. Following figures 7, 8, 9, and 10 show the simulation of the different training environments.

1.5.6 Visualization

To identify and analyze the pattern of the behaviour of data from the different fields such as business, medical, banking, etc. graphics methods are used. Nowadays supercomputers are used to process millions of data for classification based on some common behaviour. During the process, it is very difficult to observe the data pattern until and unless it represents through some diagram. The representation image may be a graph, bar diagram, pie diagram, or any other standard form. Visualization of data makes the analysis much easier and more effective. The process of producing a graphical representation of engineering, scientific, business, economics, medical, and statistical survey data is referred to as scientific visualization. The data may be 2-dimensional or 3-dimensional.



Figure 11: Illustrates a smart classroom and Simulation of high fidelity space craft modeling (Source: Internet)

The data can be scalar or vector data, any high-order tensor, or any combination of these data types.Contour plots, graphs and charts, surface renderings, and representations of volume interiors are some of the other techniques available.Image processing techniques are also integrated with computer graphics to create a variety of data visualizations, and image processing using machine learning tool enhances the process much more. E-Commerce is a trending way to increase business. To visualize the customer behaviour, the pattern of a product sales, customers interest, advertising of the new product, tracking the product delivery, etc. computer graphics are used widely and effectively. Other areas like airflow over the

surface of a spacecraft, mathematical modeling, fluid density measures through colour code, crack propagation in metals, modeling of protein, DNA, stereoscopic viewing of molecular structure, air pollution study, analysis of bowling or other activities in cricket, regeneration of different actions in different games like football, tennis ball, badminton, golf, etc., can be visualized through computer graphics attractively and effectively.



Figure 13: Structure of DNA (Source: Internet)

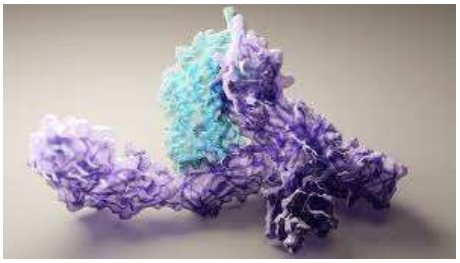


Figure 14: Structure of Protein (Source: Internet)

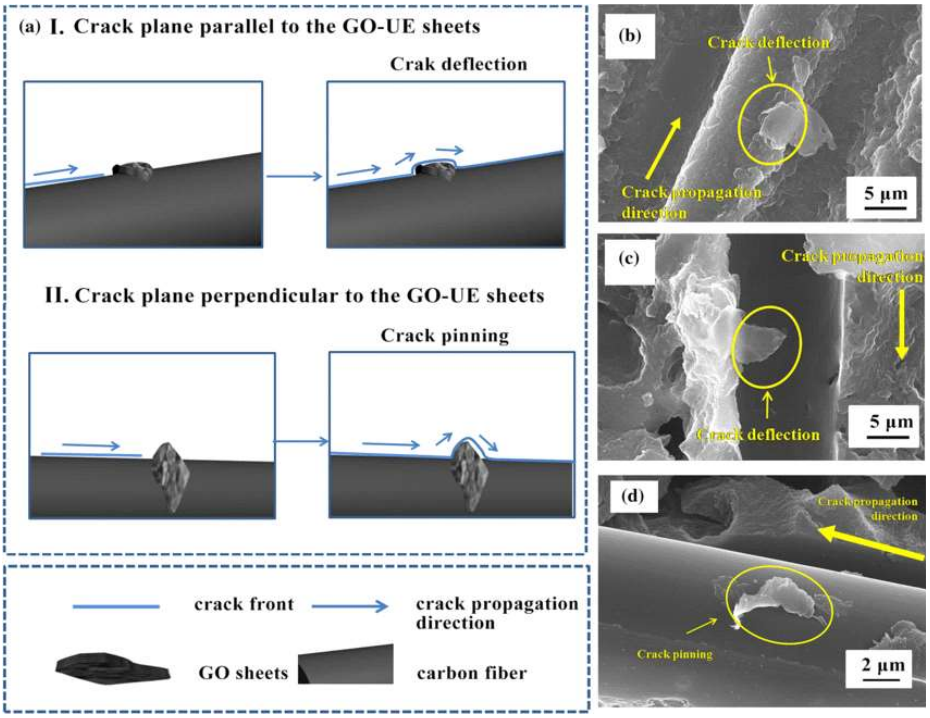


Figure 15: Illustrates Crack propagation in Metal (Source: Internet)

1.5.7 Image Processing

Image processing can be defined as a set of techniques that are used to modify or manipulate existing image definitions, whereas

computer graphics deal with image creation. Image processing can be used in improving picture quality and for the perception of visual information in the field of Artificial Intelligence. Image processing has a wide range of applications. In the field of medical science, image processing is used to analyze different types of pictures of human body parts or organs. Diagnosis of some particular diseases can be identified through image processing automatically, that is without any intervention of doctors or any experts. In the field of agriculture image processing is used to detect leaf diseases, stem diseases, bacterial infection, and soil deficiencies, also to measure PH value, nutrition amount, NPK content, and soil moisture. Proper detection and measurement of different parameters related to plant growth and fruiting can lead to healthy and beneficial farming. Nowadays through the mobile application, farmers can take a photograph of a leaf or stem and can detect the deficiencies or diseases immediately. For the wide application of image processing, drone technology is used to take photographs of gardens with big areas. Using image processing, AMAZON opened a shopping mall for customers without any salesman. Customers can pick products from stacks and add them to their carts, immediately the product details including price will reflect on the customer's mobile screen. Return of product from cart to the stack will remove the item from the list also. A set of large numbers of hidden cameras are used to detect and identify the product details with image processing techniques. Image processing is successfully used in bar code scanners, face recognition, fingerprint identification, access vehicle details from number plates, etc.



Figure 16: Depicts face recognition and leaf disease detection using image processing(Source: Internet)



Figure 17: Depicts Soil testing using image processing(Source: Internet)



Figure18: Plant health checking by a drone using image processing (Source: Internet)

1.5.8 Graphical User Interface

A graphical user interface (GUI) provides a platform to interact with the software package.The window manager is a major component of the graphical user interface that allows a user to show several windows. With graphical or non-graphical displays, each window may have a different process. To interact with a particular window, a click event is necessary to select the specific window. A good GUI should have a menu with a link for different windows or processes or different parameter values. Nowadays almost all the software packages come with GUI.



CHECK YOUR PROGRESS

Q 1: Fill in the Blanks

a) Computer graphics is primarily concerned with ……

b) In the year 1950, ……. used an electronic machine to create an image of an Oscilloscope.

c) The techniques that are used to modify or manipulate existing image definitions is known as ……

d) GUI stands for ……

1.6 INPUT DEVICES

For graphics workstations different input devices are available. Some common input devices are keyboard, mouse, joystick, image scanner, etc. some specified input devices are digitizers, trackballs, spaceballs, datagloves, touch panels, and voice systems.

A keyboard is a common input device used to provide text as input. Through the keyboard, users can input strings of characters, alphanumeric characters, numbers, operators, and different symbols. Functional keys and cursor control keys are used for performing specified predefined functions.



Figure 19: Input devices keyboard, mouse, and trackball (Source Internet)



Figure 20: Figure 19: Input devices Joystick, Image scanner, and data gloves (Source Internet)

Mouse is an input device used for pointing a point on the display screen, performing click events, and scrolling up and down a document. We can also use the mouse to zoom in and out of an image by scrolling up and down. Mouse's click event can be used to select text area, perform copy and paste, draw pictures, select menu items from GUI, etc.

Trackball and space-ball are input devices that can be rotated by the fingers or palm of the user. The rotation of the ball changes the cursor position on the screen. The trackball is a two-dimensional positioning device, whereas the space-ball provides six degrees of freedom. Spaceballs are used for three-dimensional positioning and selection operations in virtual reality systems, modeling, animation, CAD and other applications.

A joystick is a small vertical lever attached to a base that is used to move the cursor around on the screen. Some joysticks use the movement of the lever to determine position, while others use pressure on the stick.Isometric joysticks are another term for pressure-sensitive joysticks.A strain gauge measures the pressure on the joystick, which is then transformed into cursor movement in the desired direction.

Data gloves that can be used to grasp a virtual object. Hand and finger motions are detected by several sensors built into the glove. To provide information on the position and orientation of the hand, electromagnetic coupling between sending and receiving antennas is used. The glove's input can be utilized to position or manipulate virtual objects. On a video monitor, a two-dimensional projection of the scene can be viewed, or a three-dimensional projection can be viewed with a headset.

Image scanners are used to collect information regarding drawings, graphs, colour, black and white photos, text, etc. and be stored for further processing. The image data then can be used to perform transformation operations such as translation, rotation, reflection, cropping etc and fit to a required window.

Light Pens are pen or pencil size electronic devices, which can detect position on a display screen through the detection of light coming from a phosphor-coated screen. The light pens are sensitive enough to detect the light emitted from the phosphor-coated screen and detect the position. Nowadays, light pens are not used as it has lots of disadvantages compared to other input devices.

1.7 OUTPUT DEVICES

Images can have different formats. The quality of a picture depends on the size of dots, and the number of dots per inch. A printer is a hard copy output device that produces outputs either applying impact or nonimpact methods. Impact printers use an inked ribbon to print characters. A nonimpact printer may use the laser technique, ink-jet sprays, electrostatic methods, and electrothermal methods to get images. In a laser device,a laser beam causes a charge distribution on a revolving disc coated with a photoelectric material such as Selenium. The inkjet printer produces output by squirting ink in horizontal rows on a roll of paper wrapped around a drum.



Figure 21: Inkjet printer



Figure 22: Laser printer



Figure 23: Electrostatic printer

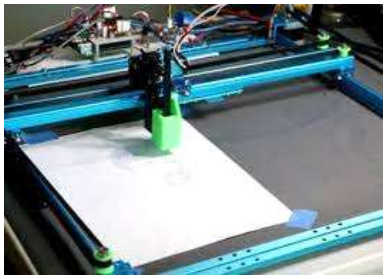


Figure 24: Plotter



1.8 LET US SUM UP

- In the era of digital science, computer graphics becomes an efficient tool for producing pictures, motion pictures, animations, and multimedia.
- An electronic circuit can be designed and developed using a simulator
- An electronic circuit can be designed and developed using a simulator
- Computer graphics can be defined as a rendering tool for the creation and manipulation of images.
- Computer graphics is the art of drawing pictures or images with the help of programming.
- Any sketch, drawing, unique artwork, or other material created with the use of a computer to pictorially describe an object or a process or transmit the information as a complement to or replacement of written descriptions is referred to as computer graphics.
- In the year 1950, BenLaposky used an electronic machine to create an image of an Oscilloscope.
- In 1961, the first video game "Space War" was launched by Steve Russel.
- PHIGS was an attempt by the ISO (International Organization for Standardization) to produce a standard computer graphics library in the 1980s.

- In computer graphics, an image or a picture consists of pixels.
- Mostly computer graphics are used in the design process such as engineering, and architectural design. Buildings, airplanes, spacecraft, textiles, watercraft, automobiles, and many other areas benefit from computer-aided design.
- Circuits related to network architecture, water supply, or any other utilities can be drawn with the repeated placement of some common graphical shapes.
- The graphics systems provide an automatic connection between the designed elements through which the designers can easily generate the required output picture.
- To identify and analyze the pattern of behaviour of data from the different fields such as business, medical, banking, etc. graphics methods are used.
- To visualize the customer behaviour, the pattern of a product sales, customers interest, advertising of the new product, tracking the product delivery, etc. computer graphics are used widely and effectively.



1.9 FURTHER READINGS

1) Book Name: COMPUTER GRAPHICS C Version, Authors: Donald D. Hearn, M. Pauline Baker, Publisher: PEARSON.
2) Book Name: Computer Graphics (Special Indian Edition), Authors: Zhigang Xiang, Publisher: Amer Media Intl - McGraw Hill audio



1.10 ANSWWERES TO CHECK YOUR PROGRESS

Ans to Q No 1:

a) image generation b) BenLaposky c) image processing d) graphical user interface.



1.11 MODEL QUESTIONS

Short Questions (Answer each question in around 150 words).

Q 1: Define computer graphics.

Q 2: Explain interactive and non-interactive computer graphics.

Q 3: List and explain the applications of computer graphics.

Q 4: State the importance of image processing.

Q 5: Differentiate between image processing and computer graphics.

Q 6: List some standard input and output device hardware used by computer graphics.

Q 7: Write in detail about digitizers.

Q 8: Write a short note on hard copy technologies.

Q 9: Explain the history of computer graphics.

Q 10: Differentiate between impact and nonimpact hard copy devices.

Q 11: Explain the use of data gloves and plotters in computer graphics.

Q 12: Describe the working principle of a joystick.

Q 13: Define the term simulator. Explain the role of a simulator in the field of education.

*** ***** ***

UNIT 2: GRAPHICS SYSTEM

UNIT STRUCTURE

2.1 Learning objectives

2.2 Introduction

2.3 Video display devices

2.4 Cathode ray tubes

2.5 Raster-scan displays

2.6 Random-scan displays

2.7 Colour - CRT Monitors

- 2.7.1 Beam penetration method
- 2.7.2 Shadow mask method

2.8 Flat panel displays

- 2.8.1 Plasma panel display device
- 2.8.2 Liquid Crystal Displays (LCD)
- 2.8.3 Electroluminescent displays

2.9 Graphics Software

- 2.9.1 GKS
- 2.9.2 PHIGS

2.10 Let Us Sum Up

2.11 Further Reading

2.12 Answer to Check Your Progress

2.13 Model Questions

2.1 LEARNING OBJECTIVES

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

- Cite the fundamentals of a graphics system
- Recognise video display devices
- Describe the working principle of CRT
- List the types of display devices
- Outline the working principle of Random and Raster scan display
- Describe how liquid crystals are used in the display device

- Recite the concept of plasma and its use in display devices.
- List different software used in computer graphics

2.2 INTRODUCTION

In the previous unit we have discussed about the definition and concept of computer graphics. Now in this unit we are going to introduce you to graphic systems. Graphics systems are used to create an image or a picture in the form of pixels drawn on the computer screen and manipulate the same. All around the world, computer graphics are used in almost all fields such as film, text documents, entertainment, photography, animation electronic gadgets, satellite imaging, scientific graphics, engineering drawings, business presentations, cartoons and artwork, simulation and modelling, GUI, etc. To enhance usability a broad range of hardware and software components are now available in the market. Two-dimensional and three-dimensional viewing is now common to almost all computers. In this chapter, we will discuss different graphics system, their features, working principle, architectures, and some of the popular and useful software packages. The next unit will be on Line drawing algorithm.

2.3 VIDEO DISPLAY DEVICES

The core hardware component of a computer graphics system is an output device. Primarily a video monitor is used by most computer graphics systems. Nowadays different types of video monitors are available such as - CRT and flat panel displays.

2.4 CATHODE-RAY TUBES

A cathode-ray tube (CRT) consists of a beam of electrons, an electron gun, control grids for controlling the deflection of the electron beam, and focused and directed the beam towards a specific location on the phosphor-coated screen of the device. When the emitted electron from the electron beam hit the phosphor-coated screen, the phosphor emits a tiny

spot of light for a moment. The light spots create a desirable picture on the screen. The light emitted by the phosphor atoms fades very rapidly. To maintain the durability of the light spot and the screen picture, some additional techniques are used. One of the popular techniques is redrawing the picture repeatedly by redirecting the electron beam on the phosphor-coated screen. The rate of regeneration or repainting of the screen picture is 75 times per second. This type of CRT is known as refresh CRT. Figure 1 depicts a typical design of a Cathode Ray Tube.

The electron gun consists of a heating filament with a heated metal cathode. When an electric current is supplied through the heating filament, it produces heat. Due to the heat generated by the filament, the negatively charged electrons of the cathode surface will be boiled off and emitted from it in a scattered manner. Applying a high positive voltage through the control grid, the scattered negatively charged electrons are directed and accelerated towards the phosphor-coated screen. The positively charged metal coating accelerator is placed within the CRT envelope near the phosphor-coated screen. In most CRTs, the control mechanism and the focusing grids are assembled within the same unit.

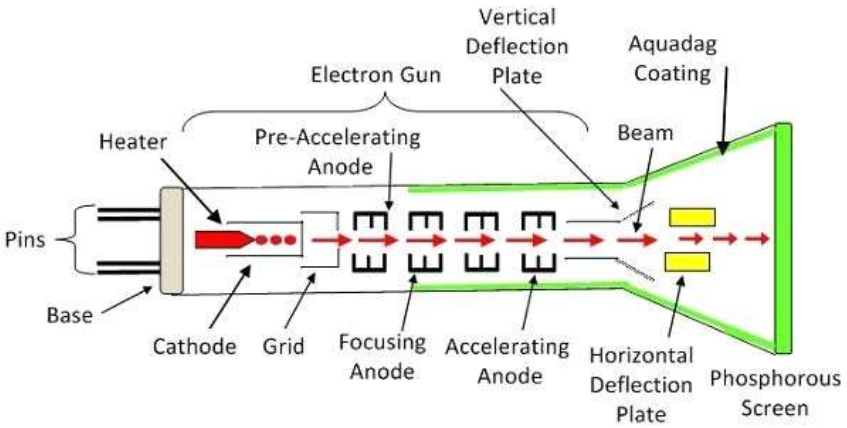


Figure 1: Cathode Ray Tube (Source: Internet)

The control grid fits over the cathode and are generally supplied with a negative voltage. By controlling the control grid voltages, the intensity of the electron beam can be controlled. The application of a high negative voltage in the control grid can repel the emitted electron from the cathode

surface and stop the electron beam from passing through the small hole at the end of the control grid towards the phosphor-coated screen. Thus by applying a low amount of negative charge on the control grid, a less number of electrons can allow passing through the hole and hit the phosphor-coated screen. The production of the light spot on the phosphor-coated screen is dependent on the number of hits made by the electrons, so by controlling the number of electrons in the beam by varying the supplied voltage into the control grid, the brightness of the display screen can be controlled. Specification of the intensity level for individual screen positions can be done through graphics software commands.

The deflection of the electron beam can be controlled by applying electric or magnetic fields. The outside of the CRT neck is mounted with magnetic deflection coils, each pair are opposite to the other. The magnetic field produced by the deflection coils results in a traverse deflection force that is perpendicular both to the direction of the magnetic field and to the direction of the travel of the electron beam. One pair of deflection coils is responsible for horizontal deflection and the other is for vertical deflection. By controlling the current supply to the deflection coil, proper deflection can be achieved.

The spot in the phosphor-coated screen is produced due to the transfer of kinetic energy of the electron beam to the phosphor. During the collision of an electron with the phosphor atom, the static phosphor atoms absorb the kinetic energy from the electron beam. A small part of the beam energy transfers to heat energy due to friction, and the remaining amount excited the outer shell electrons to move up towards the higher energy levels. But the excited or the unstable electron cannot stay at a higher energy level for a long time. After a moment it starts dropping back to its original energy band i.e. the stable ground state, by releasing the extra amount of energy in the form of a small quantum of light energy. A set of such bright spots can be visualized as a picture or an image from the front of the display screen.

For CRT, different phosphors with the colour variant are available. The persistency of the phosphor is quite good, due to which the bright spot

can be visualized after removing the electron beam also. Generally, the persistency of phosphors used in graphics systems ranges between 10 to 60 microseconds. The number of spots per centimeter that can be plotted horizontally and vertically is known as the resolution of the graphics system. For high-quality graphics, CRT has a resolution of 1280x1024. To produce equal length lines on the display screen a ratio is used known as aspect ratio, which gives the number of pixels in both the direction, vertical and horizontal. Aspect ratio 2/3 means that a vertical line plotted with two points is equal in length to a horizontal line plotted with three points.

2.5 RASTER SCAN DISPLAYS

Raster scan display is the most common type of graphics monitor used in CRTs. In a raster scan display, the electron beam is swept across the phosphor-coated screen from left to right horizontally, one line at a time, and vertically from the right end to the left end for switch into the next row as shown in figure 2. The return to the left end of the screen, after refreshing each row is known as a horizontal retraceof the electron beam. The rows are termed the scan lines. After completion of traversing each scan line, the electron beam returns to the left top corner of the screen to begin the next frame, which is known as the vertical retrace. The electron beam illuminates the dots in the desired pattern on the display screen by the switch on or off the beam intensity. The pattern or the information about the displayed image or the picture definition is stored in a memory area known as the frame buffer or refresh buffer. The frame buffer holds the intensity values for all points on the display screen. During the display of the image, the intensity of each screen point is retrieved from the refresh buffer and illuminated on the screen one scan line at a time. Each of the screen points is defined as a pixel or a picture element. The intensity or the information regarding each pixel of a screen is stored in the frame buffer. Home television systems are a good example of raster scan displays. Depending on the system capability the intensity range may differ. For a black and white screen, the intensity value will be either on or off. That is

the intensity for a pixel can be set by one bit of information, 1 for the beam intensityto be on and 0 for the beam intensity to be off. So for a bi-level system, using one bit of information the intensity of a pixel can be controlled. To achieve intensity variations or different colours on the display screen more bits can be used against each pixel definition. The frame buffer used to store bits of information regarding a black and white display device is known as a bitmap. Nowadays 24 bits per pixel are used in high-definition display devices to obtain multiple colours. The frame buffer used to store information regarding a multi-colour display device is known as a pixmap. The storage capacity for high-quality multicolour display devices can be up to several megabytes. The refreshing rate of a raster scan display ranges between 60 to 80 frames per second. The refresh rate is measured in Hz, i.e. Number of frames per clock cycle.

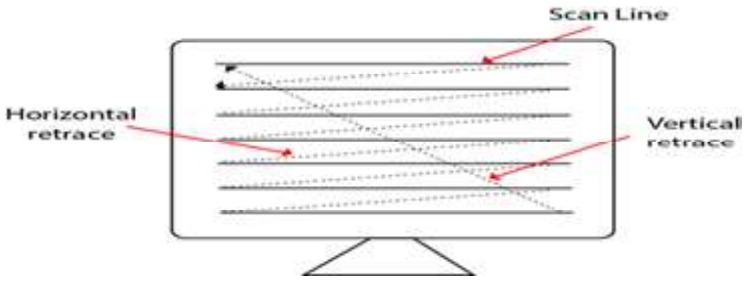


Figure 2: Scan lines in Raster scan display

2.6 RANDOM-SCAN DISPLAYS

In the case of a random-scan display, the electron beam is directed towards the part of the screen or dots on the display screen where the picture has to be drawn. Traversal of the electron beam during scan line conversion in a random scan display is shown in figure 3. The visual elements or the required picture elements are drawn one line at a time by random scan-display devices, due to which are also known as vector displays. The refresh rate depends on the number of lines to be drawn. A memory section known as the refresh display file stores the line drawing instructions or the picture definition.

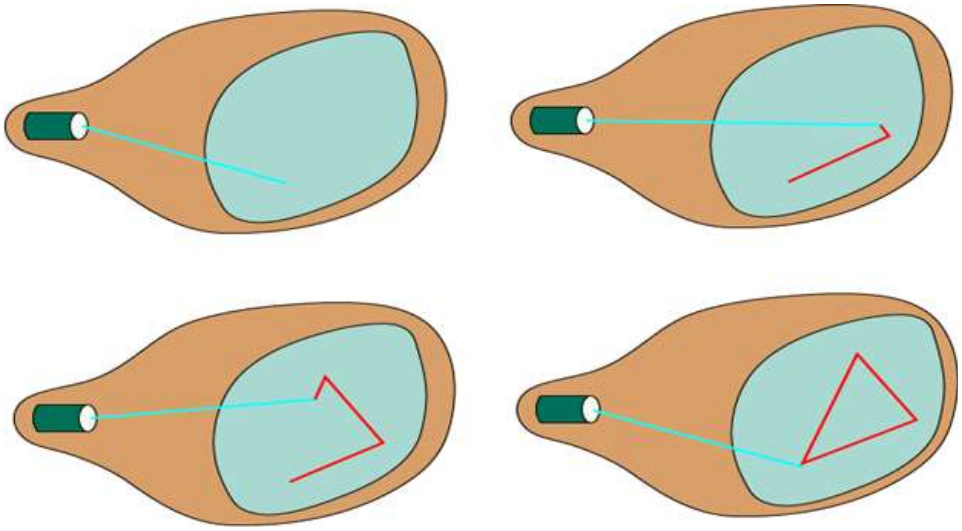


Figure 1: Random scan display (Source: Internet)

The memory area is also known as the refresh buffer. Through these specified line commands, the required picture element is drawn on the display screen. After completion of drawing all the picture elements, the electron beam returns to the first line command of the refresh buffer. This process of drawing the picture elements is repeated 30 to 60 times per second. Thus the refresh rate of the random-scan displays is 30 to 60 times per second or 30Hz to 60Hz. More than 60 times of refresh rate can lead to the burning of the phosphor atoms.

Random scan displays are better for line sketching than realistic visuals. This type of system stores the line drawing instructions rather than all points of the display screen. That is the refresh buffer does not store the intensity values for all the screen points. Thus the random-scan display or the vector display shows better resolution than a raster-scan display and produced smooth lines as the electron beam directly follows the line path. In comparison to a random scan, the raster-scan display generates jagged lines, which are plotted as discrete points.

2.7 COLOUR - CRT MONITORS

Different phosphor atoms are there, which can emit different colours. Combining more than one such phosphor atom, coloured pictures can be generated by CRT monitors. A variety of colours can be created by mixing

the emitted lights with coloured phosphor atoms.The beam penetration and the shadow mask method are the two most used ways to create coloured displays.

2.7.1 The Beam Penetration Method

To produce coloured pictures, the beam penetration method is used by the random-scan monitors. The phosphor-coated screen consists of two layers - red and green. The outer layer is red and the inner layer of phosphor coating is green. The colour of the display screen depends on the intensity of the electron beam, i.e. how far the beam can penetrate. A low-intensity electron beam can penetrate only the outer red layer, i.e. low-intensity beam can excite the phosphor atoms from the red layer. Thus a red spot can be visualized from the front of the display screen. If the electron beam accelerates toward the screen at high speed, the beam will penetrate through the red light layer and green light is emitted. For the intermediate speed of the electron beam, combinations of red and green lights are emitted to obtain multiple colours such as orange and yellow. The intensity of the electron beam can be controlled by applying different voltages at controlled grids. Thus the colour produced at different pixel positions can be controlled by the line instruction stored in the refresh buffer. The beam penetration is an inexpensive way to generate coloured pictures in random-scan monitors. Here only four colours for a pixel is possible to produce.

2.7.2 The Shadow Mask Method

Random-scan displays used the shadow-mask method to achieve multiple colours than the beam penetration method. In this method, each pixel point consists of three phosphor atoms - red, green, and blue. Each of these phosphor dots can emit light of their corresponding colour, i.e. red phosphor emits red light, the green phosphor emits green light, and the blue phosphor emits blue light.

To hit three different phosphor atoms within one pixel, three electron gun is used, one for each colour dot. Just behind the phosphor-coated screen, a shadow mask grid is used.A delta-delta shadow mask approach, which is often employed in shadow-mask systems, is depicted in figure 4.

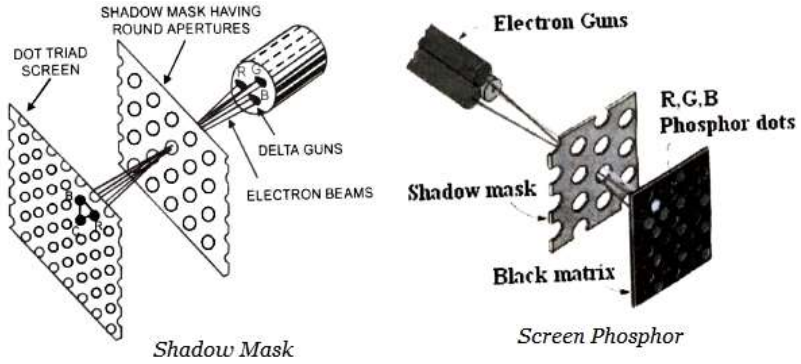


Figure 1: Depicts the structure of a shadow mask and its position (Source: Internet)

The shadow-mask grid consists of a set of holes aligned with the phosphor dots pattern in the display screen. The electron beams emitted by the three electron guns are allowed to focus on the shadow mask as a whole. The three beams trigger a dotted triangle on the display screen when they pass through a mask hole. The phosphor dots are arranged in such a way that a colour gun can activate the corresponding colour dot only. In another arrangement, the electron guns are arranged in an in-line pattern, and the three coloured phosphor atoms are also arranged in one scan line instead of in a triangular. The in-line arrangement is comparatively easy to implement and widely used in most high-resolution colour CRTs.



EXERCISE

Question1: Draw a diagram of Cathode Ray Tube and explain the functionalities of each component.

Question2: Explain the phenomenon, how light produced on a phosphor-coated screen.

Question3: Explain how colour images can display on a phosphor-coated screen.

Controlling the beam intensity level, a wide range of colour can be experienced in a shadow-mask CRT. If the switch of the blue and red gun is turned off, only the green electron gun will hit the green phosphor to produce green light on the screen. To make a small spot for each pixel position, different beam intensities can be used, because our eyes prefer to combine the three separate colours into one. By exciting the three phosphor atoms of red, green, and blue with different energy levels, different colour spots can be produced. If the three dots are excited equally, white or gray colour spots will be visible. Yellow spots can be produced by hitting the red and green phosphor dots of a pixel with equal intensities. Similarly, the equal intensity of blue and red phosphor produces a magenta colour; cyan can be produced by combining the blue and green colours. Using these three electron guns and three colours of phosphor atoms, at most eight colours can be produced. Some of the more sophisticated systems can produce millions of colours by controlling the beam intensity to some intermediate level. The control information regarding the beam intensity levels is stored in the refresh buffer. High-quality computer graphics systems use 24 bits per pixel to represent beam intensity levels by allowing 224 voltage settings for each electron gun. Using 24 bits approximately 17 million colours can be generated for a pixel. The system uses 24 bits per pixel and is known as **true-colour system** or **full-colour system.**



CHECK YOUR PROGRESS

Q 1: Define Computer graphics.

Q 2: Define Random and Raster scan displays?

Q 3: What is resolution?

Q 4: What is a frame buffer?

Q 5: What is Vertical Retrace of the Electron Beam?

2.8 FLAT PANEL DISPLAYS

CRT monitors are now replaced by some new technology such as flat-panel displays.In comparison to CRT monitors, flat-panel displays are flat and compact, require less power, and have a smaller volume and weight. Due to their thin and lightweight characteristics, flat-panel displays can hang on walls or wear them in our hands. Nowadays flat-panel displays are used in different electronic instruments such as calculators, television, smart-watch, mobile phone, laptop, different monitors, touch screen devices, etc. The flat panel displays can be categorized into two categories: emissive and non-emissive flat panel displays.

Emissive displays convert electrical energy into light energy, and in the case of non-emissive displays, optical effects were employed to turn sunshine or light from another source into graphical patterns. Some emissive displays are plasma panels, thin-film electroluminescent displays, and Light-emitting diodes (LED). Liquid crystal displays are an example of mostly used and popular non-emissive displays.

Flat CRT is also available, where the electron beams are accelerated parallelly to the display screen and deflected by 900 to the screen. Flat CRTs are not used for a long period because of heat generation and more power consumption characteristics.

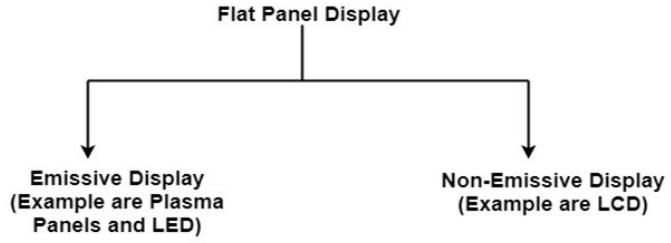


Figure 1: Categories of Flat Panel Display

2.8.1 Plasma Panel Display Device

In the plasma panel display, an amount of mixture of gas particles is sandwiched between two glass plates. Usually, neon gas is used for this purpose. Plasma panel display is also known as gas-

discharge displays because gas-particle are used to illuminate the spots on the display screen. In one glass panel, a series of conducting materials is attached vertically, and a series of conducting materials are attached horizontally in another glass panel as shown in figure 7. Applying firing voltages to these vertical and horizontal conducting ribbons, the gas-particle in the intersection region of the two conductor series can break down into glowing plasma of electrons and ions.

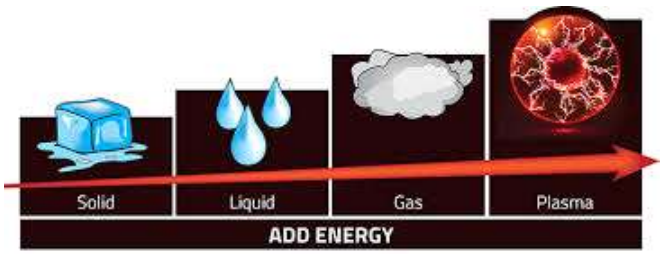


Figure 1: Different state of matter including Plasma
(Source: Internet)

Plasma can be defined as the fourth state of matter. When the gas particles are heated, it turns into plasma, which is a group of positively charged particles (ions) and negatively charged particles (electrons). Plasma is a superheated substance that has had its electrons ripped from its atoms, resulting in an ionized gas. It makes up more than 99% of the visible universe. Plasma glows in the night sky as stars, nebulas, and even auroras occasionally ripple over the north and south poles. Because plasma makes up so much of the universe, scientists from all fields are fascinated by its behaviour and qualities. Importantly, all matter is in the form of plasma at the temperatures required for viable fusion energy.

The refresh buffer stores the picture definition, and the firing voltage is used to refresh the pixel positions 60 times per second.AC voltage is employed to providea faster application of firing voltage and a brighter display.Plasma displays used to be monochromatic, however, newer systems can now display colours and grayscale as well.

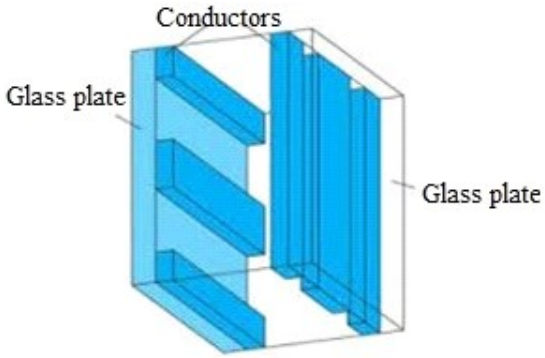


Figure 1: Design of a plasma panel display device

2.8.2 Thin-Film Electroluminescent Display

The thin-film luminescent display is also similar to the plasma panel display. The only difference is that in between the glass plates phosphor such as zinc-sulfide doped with manganese is used instead of gas to produce light spots as shown in figure 8. When sufficiently high voltages are applied to the vertical and horizontal conducting electrodes, the phosphor becomes a conductor in the area of the intersecting points of the two electrodes. When electrical energy passes through the phosphor, the manganese atoms absorbed electrical energy and jump to a higher energy state. Within a moment, the manganese atom releases the extra amount of energy in the form of light and creates the light spots which are similar to the glowing plasma. The thin-film electroluminescent displays consume more power in comparison to the plasma panel displays and good colour and greyscale are difficult to achieve.

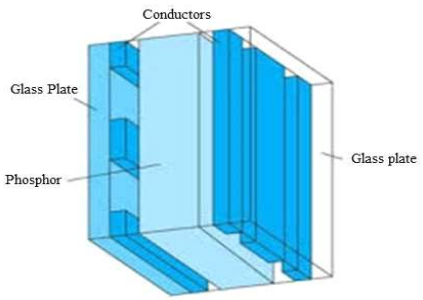


Figure 1: Basic design of thin-film electroluminescent display



EXERCISE

Question4: Draw a diagram of Plasma Panel Display and explain the functionalities of each component.

Question5: Explain the phenomenon, how light produced on a Plasma panel display.

Question6: Explain how colour images can display on a thin-film electroluminescent display.

2.8.3 Liquid-Crystal Display (LCD)

The molecules of a genuine liquid are in constant random motion; therefore their properties are uniform in all directions. The crystalline solids, in contrast, are anisotropic that is optical and other properties such as thermal and electrical conductivity vary with direction. A liquid crystal phase has many of the physical attributes of a liquid, but its molecular units are sufficiently ordered to give rise to some anisotropy, most notably in their optical properties. The properties of liquid crystals are:

- The appearance of liquid crystal phases is often hazy.
- Liquid crystals display birefringencedue to their anisotropy characteristics. That is, light entering the crystal is split into two oppositely polarised beams moving at different speeds.
- The molecules in a nematic phase (or "thread-like" crystal) are aligned in the same direction but free to move around freely, much like in a regular liquid.
- In smectic ("soap-like") phases the molecules are arranged in layers. with the long molecular axes approximately perpendicular to the laminar planes. The only long-range order extends along this axis, with the result that individual layers can slip over each other (hence the "soap-like" nature) like that observed in graphite. Within a layer, there is a certain amount of short-range order.

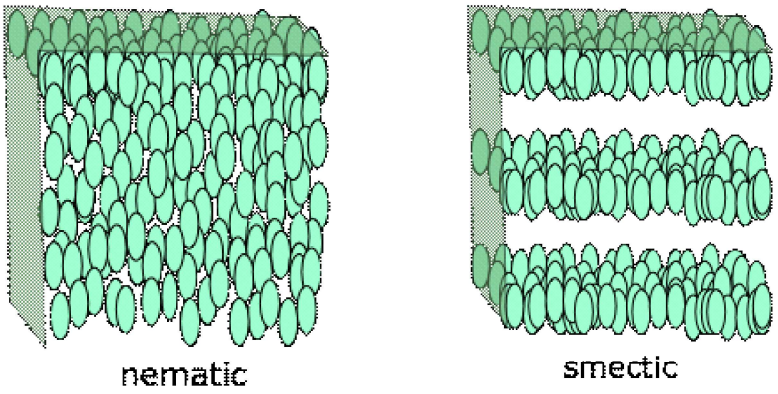


Figure 1: Illustrates the nematic and smectic structure of liquid crystal (Source: Internet)

Figure 9 illustrates the pattern of nematic and smectic liquid crystals. Unpolarized light can be defined as a light wave that is vibrating in more than one plane. The light emitted by the sun, by a lamp or a tube light are all unpolarised light sources. Polarized light can be defined as a light wave that vibrates in two dimensions only.In figure 10, it has shown that light beams coming from the source travel in a wave pattern.

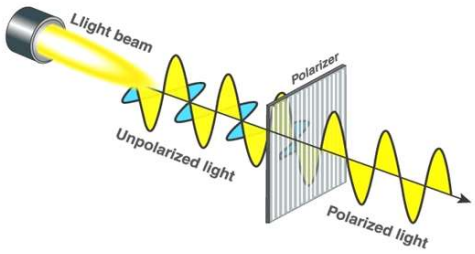


Figure 1: Illustrated the polarized and unpolarized light (Source: Internet)

The direction of the magnetic wave is directly perpendicular to the electric waves, and both the waves are perpendicular to the direction of the light beam. In the path of the light beam, a thin film or a light polarizer is placed. The polarizer reduced the dimensions of the light by stopping the wave signals travels in a particular direction. Thus the polarizer can be used as a light filter also. Liquid crystals show polarized characteristics. That is for some particular direction of the liquid crystals, it can be used as a good conductor of electricity, light, or heat. But changing the direction of the liquid crystals, the conductivity can be changed. In some particular

directions, the liquid crystals can show bad conductivity properties for light, electricity, and heat.

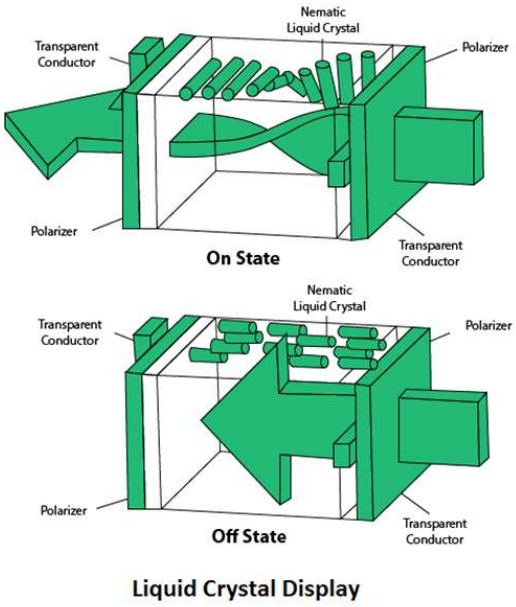


Figure 1 depicts the working principle of Liquid Crystal Display
(Source: Internet)

Suppose a horizontal polarizer can allow the horizontal light waves to pass through it, but it will stop the vertical light waves, i.e never allows the vertical light beam through a horizontal polarizer. Similarly, a vertical light polarizer allows only those light beams that vibrate in a vertical direction, and stop the horizontal waves as shown in figure 11.The term liquid-crystal refers to a crystalline arrangement of molecules that flows like liquids.Nematic or thread-like liquid crystal materials are extensively employed in LCDs. A flat panel display can be built using Nematic liquid crystal as shown in figure11. The architecture of a liquid crystal display consists of two glass plates containing light polarizers. The light polarizer of one plate is set at right angles to the other light polarizer associated with the other plate. The liquid crystal material is sandwiched between these two glass panels.One glass plate contains rows of horizontal transparent conductors, and Vertical conductor columns are inserted into the opposite plate. The pixel positions are defined by the intersection points of these two transparent conductors. The light passed through

the polarizer is twisted by the liquid crystals such that it can pass through the opposite polarizer. The state of the liquid crystals can be in two states: on or off state as shown in figure 3. The pixel spot can turn off by controlling the supplied voltage to align the liquid crystal direction, such that it cannot twist the light beam and such systems are known as passive-matrix LCD. If the light beam stops from its changing direction, it cannot pass through the polarizer. The viewer has received the light beam that has passed through the polarizer.The picture definitions are also kept in a refresh buffer, with a refresh rate of 60 frames per second.Because the light beam is generated by solid-state electricaldevices, the system is not entirely reliant on external light sources.Using thin-film transistor technology, LCDs may generate numerous colours by combining different materials or dyes and inserting a triad of colour pixels at each pixel position. In another method for constructing LCDs, transistors are employed to control the applied at each pixel position and to prevent voltage charge from leaking out of the liquid-crystal cells over time. These devices are known as active-matrix displays.



CHECK YOUR PROGRESS

Q 6. Differentiate Plasma Panel Display And Thin Film Electroluminescent Display?

Q 7. Write the merits and demerits of Plasma panel display.

Q 8. What do you mean by emissive and non-emissive displays?

Q 9. Explainthe characteristics of polarized light.

Q10. Define the term liquid crystal?

2.9 GRAPHICS SOFTWARE

2.9.1 GKS

The Graphical Kernel System (GKS) is an ISO specification that defines a standard interface for interactive computer graphics in

application programs.GKS was developed in 1977 by a group of specialists representing the national standards organizations of the world's most developed countries.The entire standard includes functional requirements for over 200 subroutines that conduct device-independent graphics input and output.Thus, application programs can easily port between different graphics devices and host computers. However, GKS is a kernel system, and is not portable. Individual GKS implementations will differ significantly since they must support various graphics devices on various systems. The layers of GKS are illustrated in figure 12. GKS never supports any graphics application software. These graphics software sit on top of the basic graphics packages. The GKS functions were created without reference to any particular programming language.In 1985, the Graphical Kernel System for two-dimensional graphics became an ISO standard. GKS functions are categorized into eight different classes such as control functions, output attributes, output primitives, segment functions, transformations, input functions, metafile functions, and inquiry functions.

Three basic parts of GKS are

- Casual presentation of the ingredients of the standards including how text is to be inserted, how polygonal zones are to be filled, and so on.
- An official description of the substances and conceptualize them in a separate functional explanation. Data like input/output parameters are contained in the functional explanations.
- The execution of abstract functions is accomplished using language bindings.

In GKS, some standard basic blocks are used to draw a picture which is known as primitives. To describe a picture, different primitives can be applied. The prime primitives used in the GKS system are:

- **Polyline:** is used to draw straight line segments providing the endpoint coordinates.

- **Poly-marker:** is used to put symbols like + * : *0*, andx against a particular coordinate.
- **Fill-area:** is used to draw polygons and fill the specified area
- **Text:** is used to add text or string at a specified coordinate.
- **Cell-array:** is used to show a colour or grayscale image at the specified location.

2.9.2 PHIGS

PHIGS (Programmer's Hierarchical Interactive Graphics Standard) is an interface between the graphics application software and the graphics subsystem. PHIGS is used to draw 2-dimensional or 3-dimensional images. It organizes visual elements in a hierarchical structure, allowing a whole assembly to be described along with all of its subassemblies. The PHIGS can be defined as a workstation that may be a single input device, a single output device or a combination of both the input and output devices, a file, a video monitor, etc.



CHECK YOUR PROGRESS

Q 11: Which of the following is a Computer Graphics type?

a) Raster and Vector
b) Raster and Scalar
c) Scalar only
d) All of the aboveQ12.List out the merits and demerits of Plasma panel display.

Q 12: Bitmap is a collection of _____ that describes an image.

a) pixels
b) algorithms
c) bits
d) colors

Q 13: Which of the following is commonly known as frame buffer on a black and white system with one bit per pixel?

a) Bitmap
b) Pix map
c) Multi map
d) All of the mentionedQ15. Define the term liquid crystal?

Q 14: What does an aspect ratio mean?

a) Ratio of vertical points to horizontal points
b) Ratio of vertical points to horizontal points and horizontal points to vertical points
c) Number of pixels
d) Ratio of horizontal points to vertical points



2.10 LET US SUM UP

- The core hardware component of a computer graphics system is an output device.
- Different types of video monitors are - CRT and flat panel displays.
- A cathode-ray tube (CRT) consists of a beam of electrons, an electron gun, control grids for controlling the deflection of the electron beam, and focused and directed the beam towardsa specific location on the phosphor-coated screen of the device.
- The refresh rate of CRT is 75 times per second.
- The electron gun consists of a heating filament with a heated metal cathode.
- Applying a high positive voltage through the control grid, the scattered negatively charged electrons are directed and accelerated towards the phosphor-coated screen.
- The control grids fit over the cathode and are generally supplied with a negative voltage.
- The production of the light spot on the phosphor-coated screen is dependent on the number of hits made by the electrons.
- The deflection of the electron beam can be controlled by applying electric or magnetic fields.

- By controlling the current supply to the deflection coil, proper deflection can be achieved.
- The persistency of phosphors used in graphics systems ranges between 10 to 60 microseconds.
- For high-quality graphics, CRT has a resolution of 1280x1024.
- To produce equal length lines on the display screen a ratio is used known as aspect ratio, which gives the number of pixels in both the direction, vertical and horizontal.
- The return to the left end of the screen, after refreshing each row is known as a horizontal retraceof the electron beam.
- The electron beam returns to the left top corner of the screen to begin the next frame, which is known as the vertical retrace.
- Picture definition is stored in a memory area known as the frame buffer or refresh buffer.
- The frame buffer used to store bits of information regarding a black and white display device is known as a bitmap.
- The frame buffer used to store information regarding a multi-colour display device is known as a pixmap.
- The visual elements or the required picture elements are drawn one line at a time by random scan-display devices, due to which are also known as vector displays.
- The refresh rate of the random-scan displays is 30 to 60 times per second
- The intensity of the electron beam can be controlled by applying different voltages at controlled grids.
- The Colour produced at different pixel positions can be controlled by the line instruction stored in the refresh buffer.
- The shadow-mask grid consists of a set of holes aligned with the phosphor dots pattern in the display screen.
- In a shadow mask method, yellow spots can be produced by hitting the red and green phosphor dots of a pixel with equal intensities.
- The flat panel displays can be categorized into two categories: emissive and non-emissive flat panel displays.

- In a Flat CRT, the electron beams are accelerated parallelly to the display screen and deflected by 900 to the screen
- In plasma panel display, AC voltage is employed to providea faster application of firing voltage and a brighter display.
- The thin-film electroluminescent displays consume more power in comparison to the plasma panel displays
- Liquid crystals show polarized characteristics
- The Graphical Kernel System (GKS) is an ISO specification that defines a standard interface for interactive computer graphics in application programs.



2.11 FURTHER READINGS

1) Book Name: COMPUTER GRAPHICS C Version, Authors: Donald D. Hearn, M. Pauline Baker, Publisher: PEARSON.
2) Book Name: Computer Graphics (Special Indian Edition), Authors: Zhigang Xiang, Publisher: Amer Media Intl - McGraw Hill audio



2.12 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: Answers to the "Check your progress"section containing multiple-choice questions

11. a
12. a
13. a
14. b
15. a



2.13 MODEL QUESTIONS

Short Questions (Answer each question in around 150 words).

Q 1: What are the raster and vector graphics?

Q 2: Write the difference between vector and raster graphics?

Q 3: What is meant by scan code?

Q 4: Explain the necessity of Scan-Line conversion in computer graphics systems.

Q 5: What is meant by refreshing the screen?

Q 6: Write the merits and demerits of Beam Penetration techniques.

Q 7: What is persistence?

Q 8: What is the Aspect ratio of a computer graphics system?

Q 9: What is a dot size?

Q 10: Define pixel.

Q 11: What is a bitmap and what is a pixmap?

Q 12: What do you mean by scan conversion?

Q 13: What is an output primitive?

Q 14: Write the two techniques for producing colour displays with A CRT?

Q 15: Explain the working principles of plasma panel display.

Q 16: Explain the working principles of Thin-Film Electroluminescent display.

Q 17: Explain the working principles of Liquid Crystal display.

Q 18: Explain how the total number of colour combinations of a pixel can be determined in a computer graphics system.

Q 19: Explain the use of GKS in a graphics system. What are the prime primitives included in GKS?

Q 20: Write a short note on computer graphics software and PHIGS.

*** ***** ***

UNIT 3: LINE DRAWING ALGORIHMS

UNIT STRUCTURE

3.1 Learning objectives

3.2 Introduction

3.3 Points and lines

3.4 Line drawing algorithms

3.5 Scan conversion of lines

3.6 DDA Line drawing algorithms

3.7 Bresenham's Line drawing algorithms

3.7.1 Steps involved in Bresenham's Line drawing algorithm

3.7.2 Advantages and disadvantages of Bresenham's Line drawing algorithm

3.8 Let Us Sum Up

3.9 Further Reading

3.10 Answer to Check Your Progress

3.11 Model Questions

3.1 LEARNING OBJECTIVES

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

- List different parameters related to graphics in a computer system.
- Explain about different issues in line drawing algorithms.
- Cite how to draw a line using DDA line drawing algorithms.
- Describe how to draw a line using Bresenham's line drawing algorithms.
- Calculate a straight line intermediate coordinates to plot pixels with examples.

3.2 INTRODUCTION

In the previous unit, we have discussed about graphic system. In this unit, you will be introduced to the concept of line drawing algorithm. The unit will first discuss about point and line and then we will discuss

about line drawing algorithm. Scan conversion of lines, DDA Line drawing algorithms and Bresenham's Line drawing algorithms will be discussed in the later part of the unit. In the next unit, we will discuss circle and ellipse drawing algorithm.

3.3 POINTS AND LINES

A straight line can be defined as the collection of dots. In coordinate geometry a straight line can be defined using the straight-line equation:

$Y = mx + c$ ---------- (1),

Where m represents the slope of the line, and c is the y-intercept. In computer graphics, line drawing can be defined as a straight line segment by plotting some discrete points between two endpoints. The line equation is used to determine the coordinates of discrete points. In computer graphics, the discrete points can be termed as PIXEL, the smallest element used to represent an object. PIXEL is the abbreviation of picture element. The computer system used binary numbers to control the display of PIXELS. For the bit value 1, the pixel becomes white, and for the value 0, the pixel turns black. By increasing the number of bits, the pixels can feel different colours. Pixel colours are determined by the number of bits used to represent it. Using two-bit binary numbers, a pixel can have three different colours: 0 0 - black, 0 1 - grey, 1 0 - grey and 1 1 - white. Number of colours can express as the power of two of the no. of bits. A system with 16 bit processor can have 216 numbers of colour combinations.

The pixel position can be represented by the two-dimensional coordinate geometry. If the coordinate of a pixel is (X, Y), then the coordinate of the next pixel in the X direction will be (X + 1, Y), and in the Y coordinate, the next position will be (X, Y + 1).

A point can be defined as a pixel in the computer graphics system. In a frame buffer system, the intensity of each pixel is calculated or measured separately. These are based on the Cartesian coordinate system. A point or a pixel is represented in terms of x and y coordinates. The total number of pixels n a display system depends on the coordinate precision and

resolution of the system. The resolution of a system or a display device can be defined as the number of pixels or dots that can be displayed in a given area of the screen. It is usually measured in dots per inch or .dpi. Greater the number of pixels infers higher the resolution. Pixels are the smallest addressable screen element. Computer graphics system images are made by controlling the intensity and colour of the pixels composing the screen. The following figure 3.1 has shown a straight line after plotting in coordinate positions. The appearance of the line is a jagged or stair-step pattern. For a system with low resolution, the appearance of straight lines or circles or any picture element will be jagged. But b increasing the number of points or resolution, the effect of stair-step can be avoided or eliminated. This is the main drawback of plotting a point in a computer graphics system. Using line drawing and circle drawing algorithms, it can be reduced.

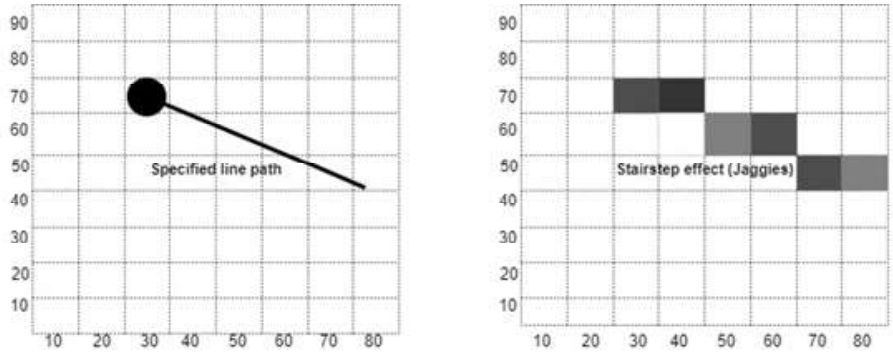
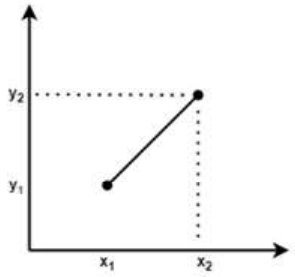


Figure 3.1 A line segment and jagged appearance of the line in graphics system

3.4 LINE DRAWING ALGORITHM

The equation of a line is shown in the expression (1). Where the slope (m) can be calculated as:



$$m = \frac{y_2 - y_1}{x_2 - x_1} \text{.....(2)}$$

$$c = y_1 - m.x_1 \text{.....(3)}$$

For any given x interval $x\Delta$ along a line, the corresponding y interval can be calculated as $\Delta y = m.\Delta x$ .....(4)

Similarly, the x interval $x\Delta$ along a line, corresponding to a specified $y\Delta$ as $\Delta x = \frac{\Delta y}{m}$.....(5)

These equations are the basis for determining the deflection voltages in analog devices.

3.5 SCAN CONVERSION OF LINES

Representation of a regular object in terms of discrete pixels is known as scan conversion. In a computer system, each graphics handler has to convert or transform the primitives into a collection of pixels. The method of scan conversion is also known as Rasterization. Each pixel in a graphics system has two states - on or off. The horizontal and vertical frequency can change for different purposes, especially in video processing, which is known as the scan converting rate.

For the scan conversion of a straight line, two endpoint coordinates are needed. In our normal life, we use scale and pencil to draw a line between two endpoints on a copy. But in a computer system, it is not possible to draw the line directly with the help of a roller. A computer can draw a line by calculating all the intermediate points in between the two endpoints. During calculation, it is possible to get fractional numbers for intermediate positions. But a computer system cannot accept a part of any pixel or a fractional number. So the fractions are rounded off to the nearest integers. Then the system fills the computed intermediate coordinate positions. A pixel never fills partially. So a straight line appearance always looks like stair-step format.

There are several algorithms used for this scan conversion. To draw a straight line - algorithms are the direct method, DDA algorithm, and Bresenham's algorithm. Similarly for drawing a circle Midpoint circle algorithm, Bresenham's Circle algorithm. To draw ellipse, midpoint ellipse drawing algorithm, etc.

Characteristics of straight lines for different slope (m):

- Slope direction can be positive, negative, zero or undefined.
- Positive slope - y values increases as the x values increases. The line slope upwards to the right.
- Negative slope - y values decreases as the x values increases. The line slope downwards to the right.
- Zero slope – y values does not change with the changing of x value. The line in exactly horizontal.
- Undefined slope – When the line exactly vertical, it does not have a definition.



CHECK YOUR PROGRESS

Q 1: Define slope of a straight line. Also explain the characteristics of straight lines for different slope.

Q 2: Explain the term scan conversion.

Q 3: Define the term Scan Conversion rate.

Q 4: Write the equation of a straight line.

3.6 DDA LINE DRAWING ALGORITHM

A digital differential analyser (DDA) is a scan conversion line algorithm based on calculating Δy or Δx. A line point is sampled in unit interval along a coordinate and determines the corresponding integer value of the line path for the other coordinate.

Consider a line with a positive slope, $(m \le 1)$. For a line with a positive slope, the gradient m is always less than or equal to one. If we sample the line path on the x-axis with the unit interval $(\Delta x = 1)$, then the successive y value can be calculated using the following equation:

$$y_{k+1} = y_k + m \quad \text{.....(6)}$$

Where the *k* values will start from 1 for the first point, and increase by 1 unit until the sample points reached the endpoint of the line. The value of *m* can range In between 0 and 1. The calculated *y* value must be rounded to the nearest integer.

For a line with a positive slope greater than 1, the role x and y will be reversed. That is the line path will be a sample in unit y intervals $(\Delta y = 1)$, and calculate each successive x value as

$$x_{k+1} = x_k + \frac{1}{m} \text{.....(7)}$$

For the equation numbers (6) and (7), it is assumed that the line path starts from left endpoints to right endpoints. If the sampling is reversed, i.e. the starting endpoint starts from the right end and ended at the left, then the value of $\Delta x = -1 \text{ and } y_{k+1} - m$.....(8)

And when the slope is greater than 1, and

$$\Delta y = -1, \text{ than } x_{k+1} - \frac{1}{m} \text{.....(9)}$$

Equations (6) and (9) can be used to calculate the line path with a negative slope also. If the slope is less than 1, and the start endpoint is at left, set $\Delta x = 1$ and calculate the successive y value with equation (6). If the start endpoint is at right and the slope is positive and the slope value is less than one, set $\Delta x = -1$ and calculate the y value using the equation (8). Similarly if the absolute value of a negative slope is greater than 1, set $\Delta y = -1$ then we can use $\Delta y = -1$ in equation (9) and $\Delta y = 1$ in equation (7). The DDA algorithm is faster than equation (1) in the calculation of pixel position. The accumulation of round-off error in the calculation of successive point coordinates can drift away from the true line path from the original one. The performance of the DDA line drawing algorithm can be improved by separating the increments m and 1/m into integer and fractional parts so that all operations are restricted within the integer operation.

DDA Algorithm:

Example 1: Draw a line segment with the endpoint coordinates (0, 0) and (6, 6) using the DDA line drawing algorithm.

Solution: The starting point coordinate of the line is (0, 0) and The right end point coordinate of the line is (6, 6)

Sl. No.	X	Y
1	0	0
2	1	1
3	2	2
4	3	3
5	4	4
6	5	5
7	6	6

So, $\Delta x = 6 - 0 = 6$ and $\Delta y = 6 - 0 = 6$, where x1=0, x2=6, y1=0 and y2=6

Thus the slope of the line m = 6/6 = 1

Since the slope of the line m = 1, the line starts at the left endpoint and is processed to the right endpoint. So the next pixel coordinate along x-axes will be: X i = X i + 1

And the corresponding y coordinate of the pixel will be calculated as yi = yi + m

The computed values for each pixel point on the line segment are shown in the table.

Example 2:Draw a line between the points (1, 1) and (8, 7) using the DDA line drawing algorithm.

Solution: The starting point is (1, 1), and the right endpoint is (8, 7)

Sl. No.	X	Y	Pixel plotted
1	2	2	2, 2
2	3	2+6/7=2.9	3, 3
3	4	2.9+6/7=3.8	4, 4
4	5	3.8+6/7=4.7	5, 5
5	6	4.7+6/7=5.6	6, 6
6	7	5.6+6/7=7.0	7, 7

The slope of the line m = (7 - 1)/ (8 - 1) = 6/7 (slope m < 1)

Here it is given that x1=1, x2=8, y1=1 and y2=7

Since the slope m < 1, the x will increment in unit intervals and y has to be determined.

So the next x coordinate position along the line path will be x1 + 1 and

And the next y coordinate of the line path will be y1 + m =y1 + 6/7 and so on. All the pixels on the line path are shown in the table:



EXERCISE

1. Draw a line between the points (12, 15) and (20, 23) using the DDA line drawing algorithm.
2. Draw a line between the points (32, 28) and (20, 18) using the DDA line drawing algorithm.

3.7 BRESENHAM'S LINE DRAWING ALGORITHM

It is the most efficient raster line-generating algorithm developed by Bresenham. In this algorithm, scan conversion is done by calculating the incremental integers which are used to display circles or any other curves. In the following Figures 4.3 and 4.4, the horizontal axes identify the pixel column and the vertical axis shows the scan line positions. While the value of x is increased by one unit, then it has to determine which of two possible pixel positions is closer to the line path at each sample step. In figure 4.3, it has to determine that, in the next pixel position whether to plot at the position (30, 30) or (30, 40). Similarly, figure3.4 shows a line with a negative slope from the left endpoint at the coordinate position (30, 70). What will be the next pixel position, whether (40, 70) or (40, 60)? In Bresenham's algorithm, it can be determined with the help of a sign of an integer parameter, which value is proportional to the difference between the separations of the two-pixel positions from the actual line path.

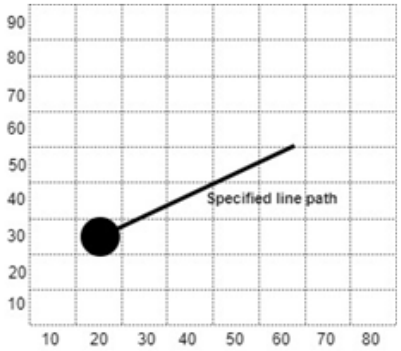


Figure 3.3 Section of a display screen Where a positive slope line is plotted

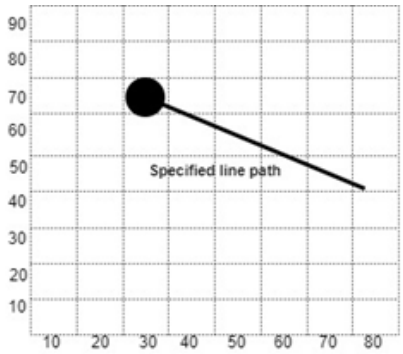


Figure 3.4 Section of a display screen Where a negative slope line is plotted

Consider a line with a positive slope of less than 1. Now, applying Bresenham's approach to determine the pixel positions along a line path by sampling at unit x intervals. Starting from the left endpoint (x_0, y_0) of a

given line, step to each Successive pixel towards x-axis, and calculates the corresponding y position which is closer to the line path. Figure 3.5 and 3.6 depicts the k^{th} step in the algorithm. Now assume that the current pixel position is determined as (x_k, y_k). Next, need to determine the y coordinate for the pixel position at $x_k + 1$. The choices for the next pixel positions are $(x_k + 1, y_k)$ and $(x_k + 1, y_k + 1)$.

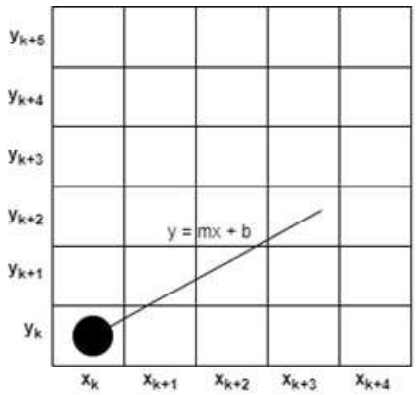


Figure 3.5 Section of a screen grid showing a pixel at (xk, yk) Position

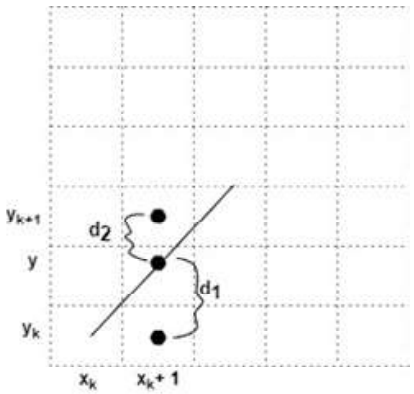


Figure3.6Distance between pixel position and theline at xk+1 position

In figure: 3.6, at the sampling position $x_k + 1$, the vertical separation from the mathematical or actual line path is considered as d1 and d2. Now the y coordinate on the actual line at sampling position $x_k + 1$ can be calculated as

$$y = m(x_k + 1) + b \text{.....(10)}$$

$$than \quad \begin{matrix} d1 = y - y_k \\ = m(x_k + 1) + b - y_k \end{matrix} \qquad and \qquad \begin{matrix} d2 = (y_k + 1) - y \\ = y_k + 1 - m(x_k + 1) - b \end{matrix}$$

The difference between these two separations is

$$\mathbf{d1 - d2 = 2\,m(x_k + 1) - 2\,y_k + 2\,b - 1 \text{.....(11)}}$$

A decision parameter pkfor the kth step in the line algorithm can be obtained from equation (11) by rearranging it. To remove the fractional part or to get an integer value, the m can be written as $\Delta y / \Delta x$, where the Δx and Δy are the vertical and horizontal separations of the endpoint positions. The pkcan be defined as:

$$d1 - d2 = 2\frac{\Delta y}{\Delta x}(x_k + 1) - 2y_k + 2b - 1$$

$$\Delta x(d1 - d2) = 2\Delta y(x_k + 1) - 2\Delta x.y_k + 2b.\Delta x - \Delta x$$

$$\therefore p_k = \Delta x(d1 - d2)$$

$$= 2\Delta y(x_k + 1) - 2\Delta x.y_k + \Delta x(2b - 1)$$

$$= 2\Delta y.x_k - 2\Delta x.y_k + 2\Delta y + \Delta x(2b - 1)$$

$$= 2\Delta y.x_k - 2\Delta x.y_k + c \quad \text{.....(12)}$$

(where c is a constant, and c=$2\Delta y + \Delta x(2b - 1)$)

The sign of the d_1- d_2 will be the same as the sign of decision parameter pk. In our example, ?x > 0, and c is constant and independent of pixel position.

If the position y_k is closer to the actual line path, then the pixel at y_k + 1, i.e. $d_1 < d_2$, then the sign of the decision parameter pk will be -ve, and the lower pixel have to be chosen. The coordinate changes occur in the unit step in either x or y directions. So the successive decision parameter p_{k+1} can have through incremental integer calculations. At step k+1, from the equation number (12), the decision parameter can be computed as:

$$p_{k+1} = 2\Delta y.x_{k+1} - 2\Delta x.y_{k+1} + c \text{.....(13)}$$

Now subtracting the equation (12) from (13) we have

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

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

Since $x_{k+1} = x_{k+1}$, the successive decision parameter for the next pixel position will be:

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

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

$$= p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k) \text{.....(14)}$$

The value of the term $y_{k+1} - y_k$ is either 0 or 1, depending on the sign of the decision parameter p_k.If y_k is closer than y_{k+1}, the sign of the decision parameter p_k is negative. At this moment we have to choose the y_k

value for the y coordinate y_{k+1}, thus the value for the expression $y_{k+1} - y_k$ = 0. If y_{k+} 1 is closer to the actual line path, the decision parameter will be positive and we have to choose the y_{k+}1 value for the coordinateposition y_k $_{+1}$. Thus the value of $y_{k+1} - y_k$ = 1 for a decision parameter with a positive sign. The calculation of decision parameters has to perform recursively to get each integer x position starting from the left end of the line path. The value of decision parameter pkfor the initial or the starting coordinates (x_0, y_0) can obtain from the equation (12)

$$p_0 = \Delta x[2m(x_0+1) + 2b - 2y_0 - 1]$$
$$= \Delta x[2(mx_0 + b - y_0) + 2m - 1]$$
$$= \Delta x.0 + 2.\frac{\Delta y}{\Delta x} - 1 \quad \text{Where } mx_0 + b - y_0 = 0 \text{ and } m = \frac{\Delta y}{\Delta x}$$
$$= 2\Delta y - \Delta x$$

3.7.1 Steps involved in Bresenham's Line drawing algorithm

1. Required two pixels coordinates as input. Consider the left endpoint coordinate as (x_0, y_0).
2. (x_0, y_0) will be the first point in the line path
3. Compute the values for the terms:
   $\Delta y, \Delta x, 2\Delta y, 2\Delta x,$ and $2\Delta y - 2\Delta x,$ and p0the decision parameter for the initial point of the line path.
4. For each pixel coordinate x_k along the line path, the following steps will perform recursively until the right endpoint. Initially k=0 or Δ x times.

4.1 if $p_k < 0$, the next point to be plot at position (x_{k+1}, y_k) and $p_{k+1} = p_k + 2\Delta y$

4.2 otherwise, the next point is to be a plot at position (x_{k+} 1, y_k $_{+1}$) and $p_{k+1} = p_{k+2}\Delta y - 2\Delta x$

Example 3: Draw a line with the endpoints (20, 10) and (30, 18), using Bresenham's line drawing algorithm.

Solution: Here it is given that the starting point coordinate is (20,

10), and the right endpoint is (30, 18).

So the $\Delta x = 30 - 20 = 10$ and $\Delta y = 18 - 10 = 8$

Initial decision parameter $p0 = 2\Delta y - \Delta x = 16 - 10 = 6$

And $2\Delta y = 16$, $2\Delta x = 20$

Therefore $2\Delta y - 2\Delta x = 16 - 20 = -4$

Initially plotting the pixel at (20, 10) and determining the successive pixel positions along the line path using the decision parameter is shown in the following table.

K	P_k	(x_{k+1}, y_{k+1})		K	P_k	(x_{k+1}, y_{k+1})
0	6	(21, 11)		5	6	(26, 15)
1	2	(22, 12)		6	2	(27, 16)
2	-2	(23, 12)		7	-2	(28, 16)
3	14	(24, 13)		8	14	(29, 17)
4	10	(25, 14)		9	10	(30, 18)

Example 4: Draw a line using Bresenham's Line algorithm with the endpoint coordinates (1, 1) and (8, 5).

Solution: Here it is given that the starting point coordinate is (1, 1), and the right endpoint is (8, 6).

So the $\Delta x = 8 - 1 = 7$ and $\Delta y = 5 - 1 = 4$

Initial decision parameter $p_0 = 2\Delta y - \Delta x = 8 - 7 = 1$

And $2\Delta y = 8$, $2\Delta x = 2$

Therefore $2\Delta y - 2\Delta x = 8 - 14 = -6$

Initially plotting the pixel at (1, 1) and determining the successive pixel positions along the line path using the decision parameter is shown in the following table.

K	P_k	(x_{k+1}, y_{k+1})		K	P_k	(x_{k+1}, y_{k+1})
0	1	(2, 2)		5	-1	(6, 4)
1	-5	(3, 2)		6	7	(7, 5)
2	3	(4, 3)		7	1	(8, 6)
3	-3	(4, 3)				
4	5	(5, 4)				

For each value of p_k, if$p_k < 0$, next point is plotted at (x_k+1, y_k), otherwise, at (x_k+1, y_k+1)



CHECK YOUR PROGRESS

Q 1: Choose the correct options for the following questions:

i. What is the formula for calculating the slope 'm' of a line?

A. m = dx / dy

B. m = dy / dx

C. y = mx + c

D. None of the above

ii. Which of the following statements is not true to the Digital Differential Analyzer (DDA) algorithm?

A. It is an incremental method of scan conversion of line

B. In this method, the calculation is performed at each step but by using the results of previous steps.

C. This algorithm can also be used to draw a circle

D. None of the above

iii. Which of the following options is not correct according to the definition of Bresenham's line drawing algorithm?

A. It gives a close approximation of points of the line by determining the n-dimensional raster that should be selected.

B. It is an incremental error algorithm.

C. It gives exact line points.

D. None of the above

iv. What is the equation for a circle?

A. y = mx + c

B. $y^2 = r^2 - x^2$

C. $rx^2y^2 = rx^2ry^2 - ry^2x^2$

D. None of the above

v. Which of the following properties is followed by the ellipse?

A. 4 - symmetry property

B. 8 - symmetry property

C. 6 - symmetry property

D. All of the above

vi. Which of the following are the methods to define a circle?

A. Using the polynomial method.

B. Using polar coordinates methods.

C. Both a. and b.

D. None of the above

Example 5: Calculate the points between the starting coordinates (9, 18) and ending coordinates (14, 22) using Bresenham's line drawing algorithm.

Solution: Given that the Starting coordinates = $(X_0, Y_0) = (9, 18)$ and the Ending coordinates = $(X_n, Y_n) = (14, 22)$

Calculate ΔX and ΔY from the given coordinates.

- $\Delta X = X_n - X0 = 14 - 9 = 5$
- $\Delta Y = Y_n - Y0 = 22 - 18 = 4$

Calculate the initial decision parameter P0

$= 2\Delta Y - \Delta X$

$= 2 \times 4 - 5$

$= 3$

So, the initial decision parameter Pk = 3

As $P_k >= 0$, Therefore,

- $P_{k+1} = P_k + 2\Delta Y - 2\Delta X = 3 + (2 \times 4) - (2 \times 5) = 1$
- $X_{k+}1 = X_k + 1 = 9 + 1 = 10$
- $Y_{k+1} = Y_k + 1 = 18 + 1 = 19$

Similarly, Step-03 is executed until the end point is reached or number of iterations equals to 4 times.

(Number of iterations = $\Delta X - 1 = 5 - 1 = 4$)

P_k	P_{k+1}	X_{k+1}	Y_{k+1}
		9	18
3	1	10	19
1	-1	11	20
-1	7	12	20
7	5	13	21
5	3	14	22

Example 6:Calculate the points between the starting coordinates (20, 10) and ending coordinates (30, 18) using Bresenham's line drawing algorithm.

Solution: Given-

- Starting coordinates = $(X_0, Y_0) = (20, 10)$
- Ending coordinates = $(X_n, Y_n) = (30, 18)$

Calculate ΔX and ΔY from the given input.

- $\Delta X = X_n - X_0 = 30 - 20 = 10$
- $\Delta Y = Y_n - Y_0 = 18 - 10 = 8$

Calculate the initial decision parameter Pk

$= 2\Delta Y - \Delta X$

$= 2 \times 8 - 10$

$= 6$

So, initial decision parameter Pk = 6

As $P_k >= 0$, Therefore

$P_{k+1} = P_k + 2\Delta Y - 2\Delta X = 6 + (2 \times 8) - (2 \times 10) = 2$ and

- $X_{k+1} = X_k + 1 = 20 + 1 = 21$
- $Y_{k+1} = Y_k + 1 = 10 + 1 = 11$

Similarly, Step-03 is executed until the end point is reached or number of iterations equals to 9 times.

(Number of iterations = $\Delta X - 1 = 10 - 1 = 9$)

P_k	P_{k+1}	X_{k+1}	Y_{k+1}
		20	10
6	2	21	11
2	-2	22	12
-2	14	23	12
14	10	24	13
10	6	25	14
6	2	26	15
2	-2	27	16
-2	14	28	16
14	10	29	17
10	6	30	18

3.7.2 Advantages and disadvantages of Bresenham's Line drawing algorithm

Bresenham's line drawing algorithm overcomes the drawbacks of the DDA line drawing algorithm. It is implemented using an incremental method for scan conversion lines than the DDA algorithm. It is faster and more efficient than the DDA line drawing algorithm. It uses only integer calculations such as addition, subtraction, and bit shifting operations to give an accurate result. The calculation for the terms 2?y - 2?x and 2?y are computed once for each line. This algorithm can be used to generate circles and different curves.

The main drawback of this algorithm is that it produces a jagged or stair-step line display.



EXERCISE

Question3: Draw a line segment using Bresenham's line drawing algorithm with the starting coordinate (30, 25) and the endpoint coordinate (10, 20).

Question4: Draw a line segment using Bresenham's line drawing line drawing algorithm with the starting coordinate (30, 25) and the endpoint coordinate (10, 20).

Question5: Draw a line segment using Bresenham's line drawing line drawing algorithm with the starting coordinate (10, 15) and the endpoint coordinate (10, 25).



3.8 LET US SUM UP

- In a graphics package, the functions which are used to describe the various picture components are known as graphics output primitives or simply primitives.
- A straight line can express using the expression y = mx + c, where m denotes the slope of the line and c is a constant.

- The slope of a line can be defined as $\Delta y / \Delta x$.
- The Digital Differential analyser (DDA) algorithm can be used to draw a straight line in computer graphics.
- DDA used an incremental approach to calculate the pixel coordinates on the line path.
- In the DDA algorithm, the round-off operation and floating-point arithmetic consume time.
- A highly accurate and efficient line drawing algorithm is Bresenham's line drawing algorithm.
- The sign of the decision parameter determines the coordinate position of the next pixel on the line path.
- Bresenham line algorithm gives a jagged or stair-steps appearance to the displayed line.
- Using the Bresenham algorithm, we can generate a circle also.



3.9 FURTHER READINGS

1) Computer Graphics, C Version, and Author: Donald D Hearn and M. Pauline Baker, Publisher: Pearson.
2) Introduction to computer graphics and multimedia, and AuthorAnirbanMukhopadhyay, Arup Chattopadhyay, Publisher: Vikas Computer graphics Multimedia and Animation, Author: Malay K. Pakhira, Publisher: PHI



3.10 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: i.) B. m = dy / dx

Ans to Q No 2: ii) C. This algorithm can also be used to draw a circle.

Ans to Q No 3: iii) C. It gives exact line points

Ans to Q No 4: iv) B. y2 = r2 - x2

Ans to Q No 5: v) A. 4 - symmetry property

Ans to Q No 6: vi) C. Both a. and b.



3.11 MODEL QUESTIONS

Short Questions (Answer each question in around 150 words).

Q 1: Answer the following questions:

a. What do you mean by graphics output primitives?

b. Explain the DDA line algorithm.

c. Write the merits and demerits of the DDA algorithm.

d. Derive the equations for Bresenham's line algorithm.

e. What are the advantages and disadvantages of Bresenham's line algorithm?

f. Explain the midpoint circle algorithm.

g. How is the decision parameter used in the midpoint circle algorithm, explain.

h. What do you mean by the slope of a line? Explain different slope properties with a suitable diagram.

i. Define ellipse? Write any two properties of an ellipse.

j. Define a circle. Also, write atleast four properties of a circle.

k. Explain the midpoint ellipse algorithm.

*** ***** ***

UNIT 4: CIRCLE AND ELLIPSE DRAWING ALGORITHM

UNIT STRUCTURE

4.1 Learning Objective

4.2 Introduction

4.3 Properties of a circle

4.4 Midpoint circle drawing algorithm

4.4.1 Steps in midpoint circle drawing algorithm

4.5 Bresenham's circle drawing algorithm

4.5.1 Steps in Bresenham's circle drawing algorithm

4.6 Ellipse generating algorithm

4.6.1 Midpoint ellipse generating algorithm

4.6.2 Steps in ellipse generating algorithm

4.7 Let Us Sum Up

4.8 Further Reading

4.9 Answer to Check Your Progress

4.10 Model Questions

4.1 LEARNING OBJECTIVES

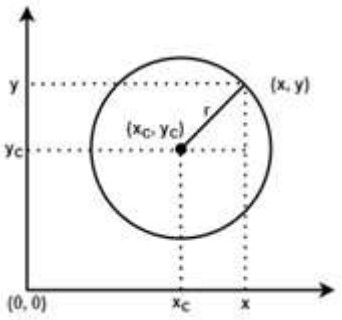
After going through this unit, you will be able to:

- describe about the properties of circle and ellipse
- identify the circle equation need to plot points on circle path.
- Recall the midpoint circle drawing algorithm
- Explain the role of decision parameter in circle drawing algorithm.
- Indicate the Bresenham's circle drawing algorithm.
- define the equations of an ellipse.

4.2 INTRODUCTION

The circle is a common picture element used in drawing pictures and graphs. A full circle or arcs of a circle can be generated through a procedure in a computer graphics system.

4.3 PROPERTIES OF A CIRCLE



A circle can be defined as a set of points that are all at a given distance r from a given point (x_c, y_c) as shown in figure 4.1. The point is known as the center of the circle and the equal distance of each point from the center to the boundary is known as the radius (r) of the circle. The complete length of the circle arc or the boundary of the circle is known as the circumference. The angle formed at the center is measured to be 360^0. A straight line from the circle boundary to the opposite boundary across the center is called the diameter. The diameter of a circle is always measured as two times the circle radius, i.e. diameter d = 2 * radius(r). The relation between radius (r) and the center (c) can be expressed in terms of Pythagorean Theorem in Cartesian coordinate as

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad \text{.....(1)}$$

Using the equation (1), it is possible to calculate the coordinate positions of circle circumference by stepping along the x-axis in unit steps from (x_c- r) to (x_c + r) and calculate the corresponding y coordinate for each pixel using the following equation (2):

$$y = y_c \pm \sqrt{r^2 - (x_c - x)^2} \quad \text{.....(2)}$$

This process of circle generation requires considerable computation at each step and the spacing between the plotted pixels is not uniform. This non-uniformity can reduce by calculating the circle boundary points using polar coordinates r and θ. The expression used to calculate the circle boundary pixel position is:

$$\left.\begin{aligned} x &= x_c + r\cos\theta \\ y &= y_c + r\sin\theta \end{aligned}\right\} \quad \text{.....(3)}$$

Another method to reduce the computation steps is the consideration of the symmetry of a circle. The shape of a circle is similar in each quadrant and each octant. We can generate the circle section in the second quadrant of the xy plane by noting that the two circle sections are symmetric to the y axis. The circle sections in the third and fourth quadrant can obtain from sections in the first and second quadrants considering symmetry to the x-axis. Circle sections in adjacent octants are symmetric about the 450 lines dividing the two octants.

Calculation of pixel positions along a circle boundary or circumference using (1) and (3) still requires a remarkable computation time. The Cartesian equation (1) is required to perform multiplication and square root operations, while equation (3) requires multiplications and trigonometric calculations. An efficient algorithm is generally based on the incremental calculation of decision parameters. One efficient algorithm to generate a circle is the Midpoint Circle Algorithm.

4.4 MIDPOINT CIRCLE DRAWING ALGORITHM

As seen in the raster line algorithm, the sampling is done at unit intervals and the closest pixel position to the specified circle path is determined at each step. Considering the radius r and the center (x_c, y_c), let us set the algorithm to calculate the circle boundary pixel positions centered at the coordinate origin (0, 0). Then each calculated position(x, y) is moved to its proper screen position by adding x_c to x and y_c to y. Along the circle section from x=0 to x=y in the first quadrant, the slope of the curve varies from 0 to -1. Therefore we can take unit steps in the positive x-direction over this octant and use a decision parameter to determine which of the two possible y positions is closer to the circle path at each step. Positions in the other seven octants can obtain by symmetry.

For applying the midpoint circle algorithm, the definition of circle function can be express as:

$$f_{circle}(x, y) = x^2 + y^2 - r^2 \quad \text{.....(4)}$$

Any point on the circle circumference with radius r satisfies the equation $f_{circle}(x, y) = 0$. For an interior point, the circle function gives a negative value. If the pixel position is outside of the circle boundary, the circle function gives a positive value. Thus checking the sign of the circle function, the position of (x, y) point can be determined. The following figure 4.2 shows the midpoint between two candidate pixels at sampling position $x_k + 1$. Assume that the currently plotted pixel is (x_k, y_k). We need to determine the next pixel position, whether the pixel position $(x_k + 1, y_k)$ or $(x_k + 1, y_k - 1)$ is closer to the circle path. The circle function from the equation (4), we can have the decision parameter to determine the midpoint between these two-pixel positions.

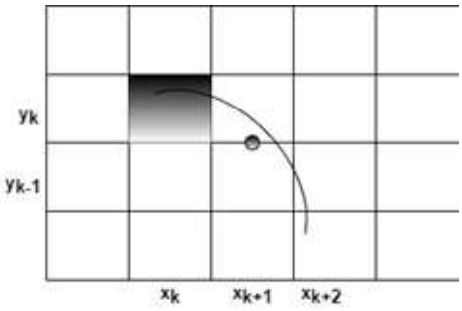


Figure 4.2 Midpoint between candidate pixels at sampling position x_k+ 1 along the circular path

$$p_k = f_{circle}(x_k + 1, y_k - \frac{1}{2})$$

$$=(x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2 \quad \text{.....(5)}$$

If pk< 0, the midpoint is inside the circle. And the pixel on scan lineyk is closer to the circle boundary.

Otherwise, the mid position is outside or on the circle boundary, and select the pixel position y_k- 1.

Successive decision parameters can be obtained using incremental calculations. We obtain a recursive expression for the next decision parameter by evaluating the circle function at sampling position $x_{k+1} + 1 = x_k + 2$:

$$p_{k+1} = f_{circle}(x_{k+1}+1, y_{k+1}-\frac{1}{2})$$

$$=[(x_k+1)+1]^2+(y_{k+1}-\frac{1}{2})^2-r^2$$

$$= p_k + 2(x_k+1)+(y^2{}_{k+1}-y^2{}_k)-(y_{k+1}-y_k)+1 \quad \ldots\ldots(6)$$

Where y_{k+1} is either y_k or y_k-1 depending on the sign of p_k.

If p_k is negative, then y_{k+1} = yk, and from the equation (6), we can calculate the successive p_{k+1}:

$$p_{k+1} = p_k + 2x_{k+1} + 1 \quad \ldots\ldots(7)$$

Otherwise, $y_{k+1} = y_k - 1$, and

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1} \quad \ldots\ldots(8)$$

Where
$$2x_{k+1} = 2x_k + 2$$
$$2y_{k+1} = 2y_k - 2$$

Initially, if the starting coordinate is (0, r), then these two terms have the value 0 and 2r respectively. Each successive value is obtained by adding 2 to the previous value of 2x and subtracting 2 from the previous value of 2y. The initial decision parameter is obtained by evaluating the circle function for the starting coordinate position (x_0, y_0) = (0, r):

$$p_0 = f_{circle}(0+1, r-1/2)$$

$$=1^2+(r-1/2)^2-r^2$$

$$=\frac{5}{4}-r \quad \ldots\ldots(9)$$

If the radius r is specified as an integer, then the value of p_0 can be calculated as:

$$p_0 = 1 - r \quad \ldots\ldots(10)$$

As in Bresenham's line drawing algorithm, the midpoint circle drawing algorithm calculates the pixel positions on circle boundary using integer addition and subtractions. The summary of the midpoint circle algorithm is as follows:

4.4.1 Steps In Midpoint Circle Algorithm:

1. Consider a circle with radius r, and center coordinate (x_c, y_c). Let the first point on the circumference of the circle centered at the coordinate origin (0, 0) is (x_0, y_0) = (0, r)
2. Calculate the initial decision parameter p_0 = 5/4 -1
3. At each x_k position, k starts from 0, perform the following operations :

   3.1 ifp_k< 0, the next point on the circle circumference will be (x_{k+1}, y_k) and $p_{k+1} = p_k + 2x_{k+1} + 1$

   Otherwise, the next point on the circle path is (x_k + 1, y_k - 1) and

   $$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

   Where
   $$2x_{k+1} = 2x_k + 2$$
   $$2y_{k+1} = 2y_k - 2$$
4. Need to determine the symmetry points for the other remaining octants.
5. Move each calculated pixel position (x, y) onto the circle boundary centered on (xc, yc) and plot the coordinate values: x = x + x_c and y = y + y_c
6. Repeat the steps from 3 to 5 until x $\leq$ y.

Example 1: Show the pixel pints on the circle boundary in the first quadrant, centered at the coordinate origin (0, 0) and the radius r = 10.

Solution: initial decision parameter p_0 = 1 - r = 1 - 10 = 9

Let the initial point of the circle is (x_0, y_0) = (0, 10)

Therefore the term $2x_0$ = 0 and $2y_0$ = 20

So the successive pixel coordinates along the circle path are:

k	P_k	(x_{k+1}, y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0	-9	(1, 10)	2	20
1	-6	(2, 10)	4	20
2	-1	(3, 10)	6	20
3	6	(4, 9)	8	18
4	-3	(5, 9)	10	18
5	8	(6, 8)	12	16
6	5	(7, 7)	14	4

Example 2: Find the points on a circle on of its octants with the circle centered at (5, 5) and has a radius of 8 units.

Solution: Assume that the circle is centered at the origin and proceed to solve. After finding the points add center (5, 5) to each point.

The initial point $(x_0, y_0) = (0, 8)$

$P_0 = 1 - r = 1 - 8 = -7$

k	P_k	at origin (0, 0) (x_{k+1}, y_{k+1})	At origin (5, 5) (x_{k+1}, y_{k+1})
0	-7	(1, 8)	(6, 13)
1	-4	(2, 8)	(7, 13)
2	1	(3, 7)	(8, 12)
3	-6	(4, 7)	(9, 12)
4	3	(5, 6)	(10, 11)
5	2	(6, 5)	(11, 10)



CHECK YOUR PROGRESS

Q 1: What is the equation for a circle?

A. $y = mx + c$

B. $y^2 = r^2 - x^2$

C. $rx^2y^2 = rx^2ry^2 - ry^2x^2$

D. None of the above

Q 2: Which of the following are the methods to define a circle?

A. Using the polynomial method.

B. Using polar coordinates methods.

C. Both A and B.

D. None of the above

4.5 BRESENHAM'S CIRCLE DRAWINGALGORITHM

Bresenham's circle drawing algorithm is successfully used to draw the circle points on its perimeter for one octant, for a given value of the circle radius (c_r) and the center (x_c, y_c) of the circle. For the first octant, the circle points will be generated from 900 to 450 as shown in the following

figure 4.3. The circle points coordinates will be transfer in the positive x (+x) axis and in the negative y (-y) axis.Bresenham's algorithm chooses the coordinate point for the circle boundary, which is closer to the circle path. In the following diagram: consider the circle radius is r.

dA represents the distance between the centre of the circle and the point A (x_k + 1, y_k), and ?Adenoted the distance of the point A from the circle boundary.

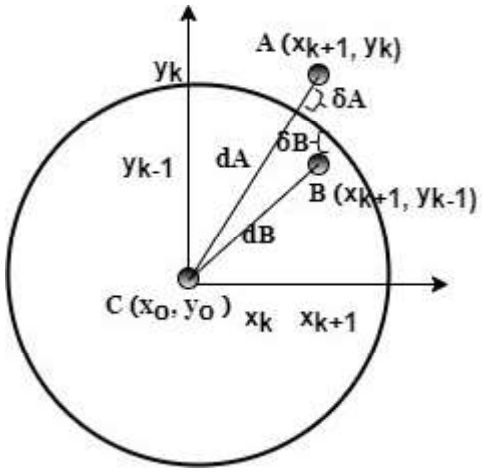


Figure 4.3: Bresemham's circle drawing algorithm

ϑB represents the distance between the centre of the circle and the point B(x_k + 1, y_k - 1), and ϑBdenoted the distance of the point B from the circle boundary.

Hence dA = r + ϑAanddB = r - ϑB

Radius (r) = dB + ϑB, now, the distance dAand dB can be computed using the following expressions:

$$dA = \sqrt{(x_k + 1 - 0)^2 + (y_k - 0)^2}$$
$$= \sqrt{(x_k + 1)^2 + (y_k)^2}$$

$$dB = \sqrt{(x_k + 1 - 0)^2 + (y_k - 1 - 0)^2}$$
$$= \sqrt{(x_k + 1)^2 + (y_k - 1)^2}$$

The ϑA is a positive value, and it can be calculated from ϑA = dA - r, similarly ϑB is a negative number, which can be calculated from the

equation ϑB = dB - r.

Now, to avoid the complex calculations containing square root, introduce two new variables as follows:

$$\delta A' = dA^2 - r^2$$
$$\delta B' = dB^2 - r^2$$

The decision parameter D can be achieved from the equation $D = \delta A' + \delta B'$, to determine the coordinate position which is closer to the circle boundary. For a positive value of D, when $\delta A' > \delta B'$, choose the coordinate point B as the next circle path point. For a negative value of D, when $\delta A' < \delta B'$, choose the coordinate point A as the next circle path point.

The decision parameter for the current pixel coordinate can be calculated as:

$$\begin{aligned} D_k &= \delta A' + \delta B' \\ &= (dA^2 - r^2) + (dB^2 - r^2) \\ &= \left[(x_k + 1)^2 - r^2 + {y_k}^2\right] + \left[(x_k + 1)^2 - r^2 + (y_k - 1)^2\right] \\ &= 2(x_k + 1)^2 + (y_k - 1)^2 + {y_k}^2 - 2r^2 \end{aligned} \quad \ldots\ldots(1)$$

The decision parameter D_{k+1} for the next pixel coordinate can be calculated from the equation (1) as following, where $x_{k+1} = x_k + 1$ and $y_{k+1} = y_k - 1$ or y_k depending on the decision parameter of the previousstep:

$$D_{k+1} = 2(x_{k+1} + 1)^2 + (y_{k+1} - 1)^2 + {y_{k+1}}^2 - 2r^2 \quad \ldots\ldots(2)$$

Now, to get the value of in terms of the previous parameter, perform subtraction operation between the equation (1) and (2).

$$D_{k+1} - D_k = 2(x_{k+1} + 1)^2 + (y_{k+1} - 1)^2 + {y_{k+1}}^2 - 2r^2 - \left[2(x_k + 1)^2 + (y_k - 1)^2 + {y_k}^2 - 2r^2\right]$$

$$\begin{aligned} \therefore D_{k+1} &= D_k + 2(x_k + 1 + 1)^2 - 2(x_k + 1)^2 + (y_{k+1} - 1)^2 - (y_k - 1)^2 + {y_{k+1}}^2 - {y_k}^2 - 2r^2 + 2r^2, \text{ here } x_{k+1} = x_k + 1 \\ &= D_k + 2(x_k + 2)^2 - 2(x_k + 1)^2 + (y_{k+1} - 1)^2 - (y_k - 1)^2 + {y_{k+1}}^2 - {y_k}^2 \\ &= D_k + 4x_k + 2({y_{k+1}}^2 - {y_k}^2) - 2(y_{k+1} - y_k)^2 + 6 \end{aligned} \quad \ldots\ldots(3)$$

If $D_k < 0$, *then* $y_{k+1} = y_k$, the equation (3) becomes

$$D_{k+1} = D_k + 4x_k + 6 \quad \ldots\ldots(4)$$

if $D_k \geq 0$, *then* $y_{k+1} = y_k - 1$, the equation (3) becomes

$$\begin{aligned} D_{k+1} &= D_k + 4x_k + 2((y_k - 1)^2 - y_k^2) - 2(y_k - 1 - y_k)^2 + 6 \\ &= D_k + 4x_k + 2(y_k^2 + 1 - 2y_k - y_k^2) + 2 + 6 \\ &= D_k + 4x_k - 4y_k + 10 \quad \text{.....(5)} \end{aligned}$$

When the value of the decision parameter $D_k < 0$, then the next pixel coordinate on the circle boundary will be $(x_k + 1, y_k)$, the decision parameter can be calculated using the equation (4).

When the value of the decision parameter $D_k \geq 0$, then the next pixel coordinate on the circle boundary will be $(x_k + 1, y_k - 1)$, the decision parameter can be calculated using the equation (5).

The initial decision parameter can be obtained from the equation (1), by placing $x_0 = 0$ *and* $y_0 = r$ the starting point of the circle path as follows:

$$\begin{aligned} D_0 &= 2(x_k + 1)^2 + (y_k - 1)^2 + y_k^2 - 2r^2 \\ &= 2(0 + 1)^2 + (r - 1)^2 + r^2 - 2r^2 \\ &= 2 + r^2 - 2r + r^2 - 2r^2 + 1 \\ &= 3 - 2r \end{aligned}$$

The initial decision parameter D_0 always gives an integral value.

4.5.1 Steps In Bresenham's Circle Drawing Algorithms:

The steps required to consider during the implementation of Bresenham's circle drawing algorithm are:

1. Determine the radius r of the circle from the given information.
2. Assume that the circle center lies on the origin (0, 0) of the given coordinate.
3. Plot the first point at $P(0, r)$, where $x = x_0$ *and* $y = r$.
4. Compute the value of the initial decision parameter by using the equation $D_0 = 3 - 2r$.
5. Repeat the steps 5 and 6 until $x \leq y$

i. *if* $d_k < 0$ *then*

$$d_{k+1} = d_k + 4x_k + 6 \text{ and}$$
$$x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k$$

ii. *if* $d_k \geq 0$ *then*

$$d_{k+1} = d_k + 4(x_k - y_k) + 10 \text{ and}$$
$$x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k - 1$$

6. Plot the new pixel (x_{k+1}, y_{k+1}).
7. Determine the pixel coordinates for the symmetric points of other octants also.
8. **Special Case:** if the center of the circle does not lie on the coordinate origin (x_0, y_0), assume that the center lies on some other coordinate (x_c, y_c), then in step 6, the coordinate of the plotted pixel will be $(x_{k+1} + x_c, y_{k+1} + y_c)$

Example 3: Calculate the circle path points for the circle with the radius r = 10 unit.

Solution: Here the center of the circle is not given, so let us assume that the center of the circle lies at coordinate origin (x_0, y_0).

Therefore, the initial decision parameter $D_0 = 3 - 2r = 3 - 2.10 = -17$

The first point on the circle path in the first octant will be

$(0, r) = (0, 10)$.

Since $d_0 < 0$, the x coordinate will increase by 1 unit and the y coordinate will remain same. So the next pixel coordinate on the circle path will be

$$x_{k+1} = 0 + 1 = 1 \quad \text{and}$$
$$y_{k+1} = y_k = 10$$
$$\therefore (x_{k+1}, y_{k+1}) = (1, 10)$$

Since the initial decision parameter has a negative value, so the $d_{k+1} = d_1$ can be computed using the equation (4).

$$d_1 = d_0 + 4x_0 + 6$$

Therefore, $= -17 + 6 \qquad \because x_0 = 0$

$$= -11$$

Since $d_1 < 0$, The x coordinate will increase by 1 unit and the y coordinate will remain same. So the next pixel coordinate on the circle path will be

$$x_{k+1} = 1 + 1 = 2 \quad \text{and}$$
$$y_{k+1} = y_k = 10$$
$$\therefore (x_{k+1}, y_{k+1}) = (2, 10)$$

Now the decision parameter for the next pixel will be:

Since the decision parameter d_1 has a negative value, so the $d_{k+1} = d_2$ can be computed using the equation (4).

$$d_2 = d_1 + 4x_1 + 6$$
$$= -11 + 4 + 6 \qquad \because x_1 = 1$$
$$= -1$$

Since $d_2 < 0$, The x coordinate will increase by 1 unit and the y coordinate will remain same. So the next pixel coordinate on the circle path will be

$$x_{k+1} = 2 + 1 = 3 \quad \text{and}$$
$$y_{k+1} = y_k = 10$$
$$\therefore (x_{k+1}, y_{k+1}) = (3, 10)$$

Again the decision parameter for the next pixel will be:

Since the decision parameter d_2 has a negative value, so the $d_{k+1} = d_3$ can be computed using the equation (4).

$$d_3 = d_2 + 4x_2 + 6$$
$$= -1 + 4.2 + 6 \qquad \because x_2 = 2$$
$$= 13$$

Since $d_3 > 0$, The x coordinate will increase by 1 unit and the y coordinate will decrease by 1 unit. So the next pixel coordinate on the circle path will be

Iteration no	x_{k+1}	y_{k+1}	d_k
0	0	10	-17
1	1	10	-11
2	2	10	-1
3	3	10	13
4	4	09	-5
5	5	09	17
6	6	08	11
7	7	07	13

Table 5: Shown he values of the example 5

$$x_{k+1} = 3+1 = 4 \quad \text{and}$$
$$y_{k+1} = y_k - 1 = 10 - 1 = 9$$
$$\therefore (x_{k+1}, y_{k+1}) = (4, 9)$$

This process will continue until $x \le y$. The output pixel coordinates for the first octant is shown in the table 5.

Example 4: Given the centre point coordinates (10, 10) and radius as 10, generate all the points to form a circle.

Solution:Given- that the Centre Coordinates of Circle (X_0, Y_0) = (10, 10) and the Radius of Circle = 10

Now consider the starting point coordinates (X_0, Y0) as-

- $X_0 = 0$
- $Y_0 = R = 10$

Calculate the value of initial decision parameter P0 as-

$P_0 = 3 - 2 \times R$

$P_0 = 3 - 2 \times 10$

$P_0 = -17$

As $P_{initial} < 0$, therefore,

- $X_{k+1} = X_k + 1 = 0 + 1 = 1$
- $Y_{k+1} = Y_k = 10$
- $P_{k+1} = P_k + 4_x X_{k+1} + 6 = -17 + (4 \times 1) + 6 = -7$

This step is applicable here as the given centre point coordinates is (10, 10).

$X_{plot} = X_c + X_0 = 1 + 10 = 11$

$Y_{plot} = Y_c + Y_0 = 10 + 10 = 20$

Continue the process until $X_{plot} => Y_{plot}$ as follows-

Algorithm calculates all the points of octant-1 of the circle and terminates.

P_k	P_{k+1}	(X_{k+1}, Y_{k+1})	(X_{plot}, Y_{plot})
		(0, 10)	(10, 20)
-17	-7	(1, 10)	(11, 20)
-7	7	(2, 10)	(12, 20)
7	-7	(3, 9)	(13, 19)
-7	15	(4, 9)	(14, 19)
15	13	(5, 8)	(15, 18)
13	19	(6, 7)	(16, 17)
Algorithm Terminates			

EXERCISE

Question1: Show the set of points of a circle boundary in first quadrant using Midpoint circle drawing algorithm with the center coordinate (3, 3) and the radius r= 12.

Question2: Show the set of points of a circle boundary in first quadrant using Midpoint circle drawing algorithm with the center coordinate (0, 0) and the radius r= 7.

4.6 ELLIPSE GENERATING ALGORITHM

Ellipse can be defined as a set of points such that the sum of the distances from two fixed positions known as foci is the same for all points. If the distances between the point p = (x, y) to the foci are d1 and d2 then the general equation of an ellipse can be stated as:

d1 + d2 =constant …… (24)

If the coordinates of the foci are f1(x_1, x_2) and f2(x_2, y_2), then the equation becomes

$$\sqrt{(x-x_1)^2+(y-y_1)^2}+\sqrt{x-x_2)^2+(y-y_2)^2}=cons\tan t.....(25)$$

The major axis of an ellipse is the straight line from one end of the ellipse boundary to the other across the foci. A straight line from one end of the ellipse at the shortest diameter part to the other end, intersecting the major axis at the middle is known as the minor axis. The intersecting point is known as the ellipse center between the two foci as shown figure 4.4.

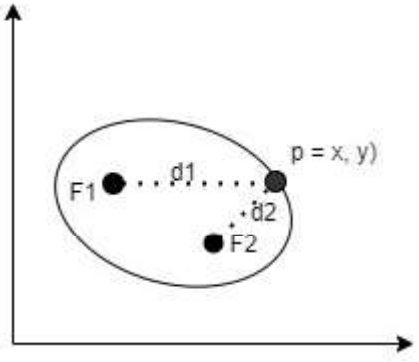


Figure 4.4Ellipse at foci F1 and F2

The ellipse equation can be expressed in terms of major and minor axes. It becomes more simplified if the axes' orientation is aligned with the coordinate axes. The following figure 4.5, it has been shown a standard position of an ellipse; where the major and minor axes are aligned parallel to the x and y axes. The semi-major and semi-minor axes are represented by the labels rx and ry. The ellipse equation can be written in terms of center coordinates (xc, yc), semi-major and minor axes is as follows:

$$\left(\frac{x-x_c}{r_x}\right)^2+\left(\frac{y-y_c}{r_y}\right)^2=1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(24)$$

Using polar coordinates r and θ, the equation of an ellipse can be expressed as:

$$\left.\begin{aligned} x &= x_c + r_x\cos\theta \\ y &= y_c + r_y\sin\theta \end{aligned}\right\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(25)$$

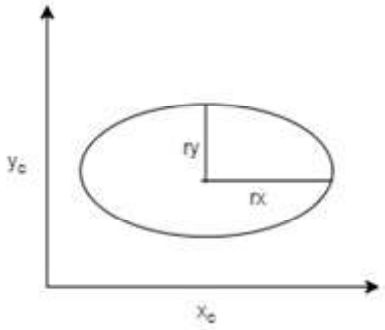


Figure 4.5Ellipse centered at (xc, yc), with semi major axis rx and semi minor axis ry.



CHECK YOUR PROGRESS

iii. Which of the following properties is followed by the ellipse?

A. 4 - symmetry property
B. 8 - symmetry property
C. 6 - symmetry property
D. All of the above

iv. The value of the decision parameter determines whether the mid-point lays ___ boundary and the then position of the mid-point help in drawing the ellipse.

A. At boundaries
B. Inside, outside, or on the ellipse
C. Only inside the boundaries
D. Only outside the boundaries

Using symmetry concepts, the steps required in computations can be reduced. An ellipse does not show the symmetric property in between octants. It shows symmetric property only on quadrants. So during the computation of boundary path pixel coordinates, need to calculate the pixel positions along the elliptical arc across one quadrant. Then the position of the pixels in the other quadrant can be calculated using symmetry.

4.6.1 Midpoint Ellipse Generating Algorithm

The goal of this algorithm is to display an ellipse. With the given parameters r_x and r_y and center coordinates (x_c, y_c) we have to determine the pixel coordinates (x, y) for an ellipse in a standard position centered at the coordinate origin and finally shift to the points such that the ellipse centered at (x_c, y_c). To display the ellipse in a non-standard position, the ellipse has to rotate about its center coordinates to re-orient the major and minor axes. Here we will consider the only standard position to display the ellipse.

The Midpoint ellipse algorithm is applied throughout the first quadrant in two parts. Following figure 4.6 shows the division of the first quadrant according to the slope of an ellipse with $r_x < r_y$. We have to process this quadrant by taking unit steps in the x - the direction where the slope of the curve has a magnitude less than 1 and taking unit steps in the y-direction where the slope has a magnitude greater than 1. In figure 4.7, we can start our plotting from the point $(0, r_y)$ and step clockwise direction along the elliptical path in the first quadrant, shifting from unit steps in x to unit steps in

y when the slope becomes less than -1.

Now consider the ellipse function derived from the equation (24) with $(x_c, y_c) = (0, 0)$ as:

$$f_{ellipse}(x,y) = r^2_y x^2 + r^2_x y^2 - r^2_x r^2_y \quad \text{.....(26)}$$

This has the following properties:

$$f_{ellipse}(x,y)\begin{cases} < 0, if\ (x,y)\ is\ inside\ the\ ellipse\ boundary \\ = 0, if\ (x,y)\ is\ on\ the\ ellipse\ boundary \\ > 0, if\ (x,y)\ is\ outside\ the\ ellipse\ boundary \end{cases} \quad \text{.....(27)}$$

Thus the ellipse function can be used as a decision parameter in the midpoint algorithm to generate an ellipse. At each sampling position, the next pixel position along the ellipse boundary can be determined using the sign of the decision parameter or the ellipse function, evaluated at the midpoint between the two candidate pixels.

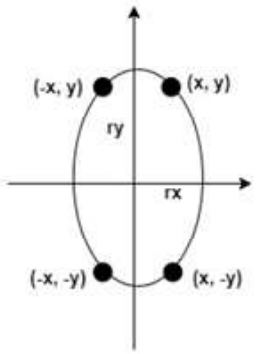


Figure 4.6Symmetry of an ellipse

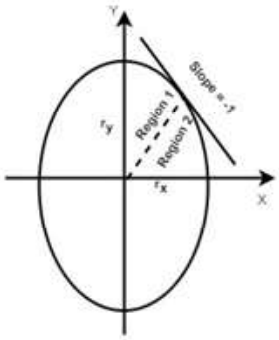


Figure 4.7Ellipse processing regions

Starting at $(0, r_y)$ we take unit steps in the x-direction until we reach the boundary between the region 1 and 2 as shown in the following figure 4.7. Then we have to shift the unit steps in the y-direction over the remainder of the curve in the first quadrant. At each step, we need to test the value of the slope of the curve. The ellipse slope can be calculated as:

$$\frac{dy}{dx} = -\frac{2r^2_y x}{2r^2_x y} \quad \text{.....(28)}$$

At the boundary between region 1 and region 2, dy/dx = - 1 and

$$2r^2_y x = 2r^2_x y$$

Therefore we have to move out from region 1 when,

$$2r^2_y x \geq 2r^2_x y$$

Figure 4.8 shows the midpoint between the two candidate pixels at sampling position x_k+1 in the first region. Assume that the point selected in the previous step is (x_k, y_k), and now we have to determine the next position along the ellipse path by evaluating the decision parameter at this midpoint.

$$p1_k = f_{ellipse}(x_k + 1, y_k - \tfrac{1}{2})$$
$$= r^2_y (x_k + 1)^2 + r^2_x (y_k - \tfrac{1}{2})^2 - r^2_x r^2_y \quad \text{.....(29)}$$

If $p1_k < 0$, the midpoint is inside the ellipse boundary and the pixel on scan line y_k is closer to the ellipse boundary. Otherwise, the midpoint will remain on or outside the ellipse and the point on scan line y_k - 1 will be closer to the ellipse boundary.

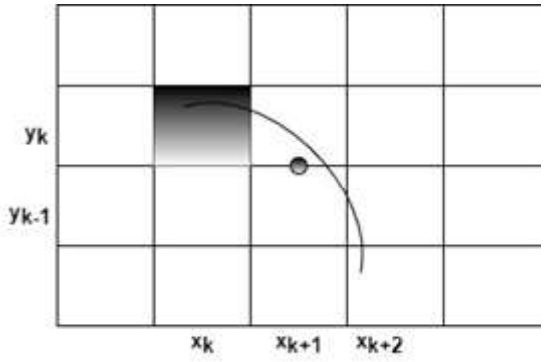


Figure 4.8Midpoint between candidate pixels at xk+1 position along an ellipse line path

At the next sampling position ($x_{k+1} + 1 = x_k + 2$), the decision parameter for region 1 is evaluated as:

$$p1_{k+1} = f_{ellipse}(x_{k+1} + 1, y_{k+1} - \tfrac{1}{2})$$
$$= r^2_y [(x_k + 1) + 1]^2 + r^2_x (y_{k+1} - \tfrac{1}{2})^2 - r^2_x r^2_y$$
$$= p1_k + 2r^2_y (x_k + 1) + r^2_y + r^2_x \left[\left(y_{k+1} - \tfrac{1}{2} \right)^2 - \left(y_k - \tfrac{1}{2} \right)^2 \right]$$

Where y_{k+1} is either y_k or $y_k - 1$, depending on the sign of decision parameter $p1_k$.

If $p1_k < 0$, then y_{k+1} is y_k, so the next point along the ellipse centered on (0, 0) is (x_{k+1}, yk)

$$p1_{k+1} = p1_k + 2r^2_y x_{k+1} + r^2_y$$

Otherwise, the next point along the ellipse is $(x_k + 1, y_k - 1)$

$$p1_{k+1} = p1_k + 2r^2_y x_{k+1} - 2r^2_x y_{k+1} + r^2_y$$

Decision parameters are incremented by the following amounts:

$$increment = \begin{cases} 2r^2_y x_{k+1} + r^2_y, & if\ p1_k < 0 \\ 2r^2_y x_{k+1} + r^2_y - 2r^2_x y_{k+1}, & if\ p1_k \geq 0 \end{cases}$$

In region 1, the initial value of the decision parameter is obtained by evaluating the ellipse function at the start position $(x_0, y_0) = (0, r_y)$;

$$\begin{aligned} p1_0 &= f_{ellipse}(1, r_y - 1/2) \\ &= r^2_y + r^2_x (r_y - 1/2)^2 - r^2_x r^2_y \\ &= r^2_y - r^2_x r_y + 1/4\, r^2_x \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (30) \end{aligned}$$

In region 2, we sample at unit steps in the negative y-direction, and the midpoint is now taken between horizontal pixels at each step. For this region the decision parameter is evaluated as:

$$\begin{aligned} p2_k &= f_{ellipse}(x_k + 1/2, y_k - 1) \\ &= r^2_y (x_k + 1/2)^2 + r^2_x (y_k - 1)^2 - r^2_x r^2_y \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (32) \end{aligned}$$

If $p2_k > 0$, the mid position is outside the ellipse boundary and we have to select the pixel x_k.

If $p2_k \leq 0$, the mid position is inside or on the ellipse boundary and we have to select the pixel $x_k + 1$.

4.6.2 Steps Involved In Midpoint Ellipse Algorithm

1. Input r_x, r_y, and the ellipse center (x_c, y_c) and obtain the first point on an ellipse centered on the origin as $(x_0, y_0)= (0, r_y)$.
2. Calculate the initial value of the decision parameter in region 1 as:

$$p1_0 = r^2_y - r^2_x r_y + \frac{1}{4} r^2_x$$

3. At each xk position in region 1, starting at k=0, perform the following test:

   If $p1_k < 0$, then y_{k+1} is y_k, so the next point along the ellipse centered on (0, 0) is (x_{k+1}, y_k)

$$p1_{k+1} = p1_k + 2r^2_y x_{k+1} + r^2_y$$

   Otherwise, the next point along the ellipse is $(x_k + 1, y_k - 1)$

$$p1_{k+1} = p1_k + 2r^2_y x_{k+1} - 2r^2_x y_{k+1} + r^2_y$$

   And continue until $2r^2_y x \geq 2r^2_x y$

4. In region 2, calculate the initial value of the decision parameter using the last point (x_0, y_0) calculated in region 1 as::

$$p2_0 = r^2_y (x_0 + \frac{1}{2})^2 + r^2_x (y_0 - 1)^2 - r^2_x r^2_y$$

5. At each yk position in region 2, starting at k=0, perform the following test:

$$p2_{k+1} = p2_k + 2r^2_y x_{k+1} - 2r^2_x y_{k+1} + r^2_x$$

   Using the same incremental calculations for x and y as in region1.
6. Determine the symmetry points in the other three quadrants.
7. Move each calculated pixel position (x, y) onto the elliptical path centered on (x_c, y_c) and plot the coordinate values: $x = x + x_c$ $y = y + y_c$
8. Repeat the steps for region 1 until $2r^2_y x \geq 2r^2_x y$

Example 5: Using mid pointellipse algorithm, solve for $r_x = 8$ and $r_y = 6$.

Solution: Given that the $r_x = 8$ and $r_y = 6$.

Assume that the ellipse centre lies on(x_c, y_c)

For region 1:

The first point on an ellipse centered on the origin (x_0, y_0) = (0, r_y) = (0, 6)

The initial value of the decision parameter is as:

$$d_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2 = 6^2 - 8^2.6 + \frac{1}{4}.8^2 = -332$$

For i=0, d_0= -332 < 0, then by the midpoint ellipse algorithm, the next point along the ellipse centered on (0, 0) is

$$(x_i + 1, y_i) = (x_0 + 1, y_0) = (1, 6)$$

$$d_1 = d_0 + 2r_y^2 x_1 + r_y^2 = -332 + 2.6^2.1 + 6^2 = -332 + 72 + 36 = -224$$

For i=1, d_1= -224< 0, the next point along the ellipse centered on (0, 0) is

$$(x_i + 1, y_i) = (x_2, y_2) = (x_1 + 1, y_1) = (2, 6) \text{ and}$$

$$d_2 = d_1 + 2r_y^2 x_2 + r_y^2 = -224 + 2.6^2.2 + 6^2 = -224 + 144 + 36 = -44$$

For i=2, d_2= - 44< 0, the next point along the ellipse centered on (0, 0) is

$$(x_i + 1, y_i) = (x_3, y_3) = (x_2 + 1, y_2) = (3, 6) \text{ and}$$

$$d_3 = d_2 + 2r_y^2 x_3 + r_y^2 = -44 + 2.6^2.3 + 6^2 = -44 + 216 + 36 = 208$$

For i=23, d_3 = 208 > 0, the next point along the ellipse centered on (0, 0) is

$$(x_i + 1, y_i - 1) = (x_4, y_4) = (x_3 + 1, y_3 - 1) = (4, 5) \text{ and}$$

$$d_4 = d_3 + 2r_y^2 x_4 - 2r_x^2 y_4 + r_y^2 = 208 + 2.6^2.4 - 2.8^2.5 + 6^2 = 208 + 288 - 640 + 36 = -108$$

Similarly, the successive decision parameter values and positions along the ellipse path are calculated uing these algorithm are:

i	d_i	(x_{i+1}, y_{i+1})
0	-332	(1, 6)
1	-224	(2, 6)
2	-44	(3, 6)
3	208	(4, 5)
4	-108	(5, 5)
5	288	(6, 4)
6	244	(7, 3)

For region 2:

The first point on an ellipse centered on the origin $(x_0, y_0) = (0, r_y) = (7, 3)$

The initial value of the decision parameter is as:

$$d_0 = r_y^2\left(x_0 + \frac{1}{2}\right)^2 + r_x^2(y_0 - 1)^2 - r_x^2 r_y^2$$

$$= 6^2\left(7 + \frac{1}{2}\right)^2 + 8^2(3-1)^2 - 8^2 6^2$$

$$= -23$$

For i=0, d_0= -23< 0, then by the midpoint ellipse algorithm, the next point along the ellipse centered on (0, 0) is

$$(x_i + 1, y_i - 1) = (x_1, y_1) = (x_0 + 1, y_0 - 1) = (8, 2)$$

$$d_1 = d_0 + 2r_y^2 x_1 - 2r_x^2 y_1 + r_x^2 = 361$$

For i=1, d_1= 361> 0, then by the midpoint ellipse algorithm, the next point along the ellipse centered on (0, 0) is

$$(x_i, y_i - 1) = (x_2, y_2) = (x_1, y_1 - 1) = (8, 1)$$

$$d_2 = d_1 - 2r_x^2 y_2 + r_x^2 = 297$$

For i=2, d_2= 297> 0, then by the midpoint ellipse algorithm, the next point along the ellipse centered on (0, 0) is

$$(x_i, y_i - 1) = (x_3, y_3) = (x_2, y_2 - 1) = (8, 0)$$

$$d_3 = d_2 - 2r_x^2 y_3 + r_x^2 = 361$$

Similarly, the successive decision parameter values and positions along the ellipse path are calculated uing these algorithm are:

i	d_i	(x_{i+1}, y_{i+1})
0	-23	(8, 2)
1	361	(8, 1)
2	297	(8, 0)



EXERCISE

Question3: Using midpoint ellipse algorithm, draw the boundary points of an ellipse with $r_x = 8$ and $r_y = 6$.

Question4: Using midpoint ellipse algorithm, draw the boundary points of an ellipse with $r_x = 8$ and $r_y = 6$.



4.7 LET US SUM UP

- In a graphics package, the functions which are used to describe the various picture components are known as graphics output primitives or simply primitives.
- One of the most used and efficient circles generating algorithms is the Midpoint Circle algorithm.
- An ellipse can be defined as a set of points located at a distance from two fixed points, where the sum of the distances from each point to the boundary points is equal. The specific points are known as the foci of the ellipse.
- The major axis of an ellipse connects both the end of the ellipse across the foci.
- The minor is a straight line segment drawn at the middle of the narrowest part of an ellipse by intersecting the major axes. The intersecting point is known as the center of the ellipse.
- The circle can divide into eight symmetric segments.
- An ellipse can divide into four symmetric segments.



4.8 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1:

i. B. y2 = r2 - x2

ii. C. Both A and B.

iii. A. 4 - symmetry property

iv. B. Inside, outside, or on the ellipse



4.9 MODEL QUESTIONS

Short Questions (Answer each question in around 150 words).

a. What do you mean by graphics output primitives?

b. Explain the midpoint circle algorithm.

c. How is the decision parameter used in the midpoint circle algorithm, explain.

d. Define the term ellipse? Write any two properties of an ellipse.

e. Define a circle. Also, write at least four properties of a circle.

f. Explain the midpoint ellipse algorithm.

*** ***** ***

UNIT 5: FILLED AREA ALGORIHMS

UNIT STRUCTURE

5.1 Learning Objectives

5.2 Introduction

5.3 Scan line polygon fill algorithm

5.3.1 Global edge table

5.3.2 Active edge table

5.4 Inside outside test

5.4.1 Odd-Even rule

5.4.2 Non-Zero winding number rule

5.5 Boundary fill

5.5.1 4-Connected area fill algorithm

5.5.2 8-Connected area fill algorithm

5.6 Flood fill algorithm

5.7 Let Us Sum Up

5.8 Further Reading

5.9 Answer to Check Your Progress

5.10 Model Questions

5.1 LEARNING OBJECTIVES

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

- How to fill an area of an object.
- What is area filled algorithm.
- How a scan line is used in area filled algorithm.
- Scan line polygon filling algorithm.
- How to detect a point whether it is an interior or exterior to the polygon.
- Learn the working principle of the odd-even rule used to identify interior points of a polygon.
- Working principle of winding number rule used in the inside-outside test.

- How does the boundary fill algorithm fill the interior points of a polygon with a specified colour?
- Working principle of the 4-connected and 8-connected approach used in the boundary-filled algorithm.
- Drawbacks of the boundary fill algorithm.
- Flood fill algorithm.
- How flood fill algorithm overcomes the drawbacks of the boundary fill algorithm.

5.2 INTRODUCTION

Output primitives of most of the graphics packages consist of solid colour or patterned polygon areas. Due to the linear boundaries of a polygon, polygon area primitives are easier than other alternatives. Filling a polygonal area can be done by calculating the overlap intervals of scan lines that traverse it. Another technique to fill an area is to start at a specific place within a defined interior region and fill colour to the surrounding pixels until a specified boundary line or conditions are found. The scan line method is used to fill simple regions such as circles, rectangles, polygons, ellipses, and other simple curve areas in basic graphics software. Sophisticated graphics packages and interactive painting systems commonly use the fill method from an interior location to determine more complex borders.

5.3 SCAN-LINE POLYGON FILL ALGORITHM

Scan-line polygon filling algorithm is used for the solid filling of polygon areas. Figure 1 shows some scan lines passing through a polygon area. Each horizontal scan line intersects polygon edges at some pixel coordinates. The polygon fill algorithm locates the intersecting points of the scan lines that make with the polygon edges. Considering the scan lines from top to bottom order, intersecting point coordinates are arranged in ascending order concerning the x coordinates i.e. from left to right and make pair of intersecting points.

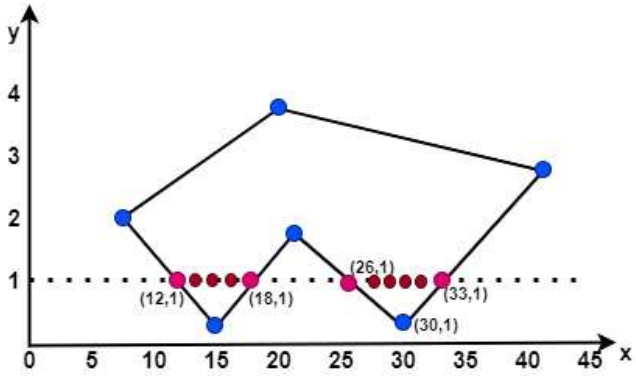


Figure 1Interiorpixels along a scan line passing through polygon edges

The corresponding frame buffer positions between the pairs will be filled with specified colours. Figure 1 shows a dotted scan line across the polygon area. The scan line intersects the polygon edges at coordinates (12, 1), (18,1), (26, 1), and (33, 1), which are arranged in sorted order on the value of the x-coordinate. The pair of consecutive coordinates are (12, 1), (18, 1), and (26, 1), (33, 1).

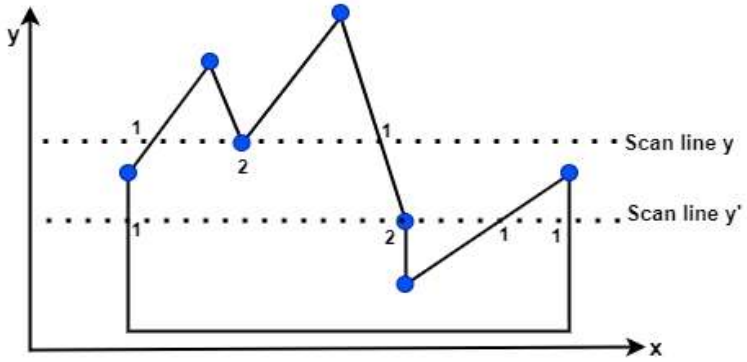


Figure 1 Intersection points at polygon edgeswith common vertices

The pixel positions between the pair coordinates are now can be filled with solid colours or any specified colour. A scan line as shown in figure 1 can identify the interior pixels correctly. But for some cases, it is required to perform some additional processing is to determine the correct interior points. In the following figure 2, the scan lines y and y' pass through the polygon edges. For the scan line y', two intersecting edges share a common vertex and the polygon edges are in opposite directions. For the scan line y, two intersecting edges share a common point, where both the edges are lies in the same direction, i.e. both are above the scan line. For the common

vertex of the polygon edges, the pairing of pixel points will not work properly, and thus scan line filling cannot be completed successfully. Thus some additional processing steps need to be performed for those that have connecting edges on the opposite side of the scan line. The existence of the connecting edges on opposite sides of a scan line can be identified by traversing the polygon boundary in a clockwise or anticlockwise direction. By observing the y coordinate of each endpoint of polygon edges, these connected vertices can be identified. If the y coordinate of two consecutive endpoints is monotonically decreased or increased, then the middle vertex needs to be counted as a single intersection point for any scan line passing through it. When the scan line found an intersecting point it starts its counting and it will be continued up to the last intersecting point. For odd count, the polygon fill algorithm starts to fill the pixels until the count becomes an even number. From the even count to the next odd count, all pixels will be considered exterior points. In the connecting point where both the polygon edges are lies in the same direction, the count will be considered two times consecutively. In figure 2, the scan liney intersects the first polygon edge at point 1 and set the counter value 1. The algorithm starts to fill the colour across the scan line up to the next intersecting point. The next intersecting point 2 in the scan line is a connecting point where two polygon edges are connected and lie on the same side of the scan line. So the counter will increase its value twice consecutively, i.e. for the counter value 2, the algorithm will stop its colour fill activity. But for the next increment of the counter value is an odd number, so the polygon fill will start its colour to the next even number of the counter. Thus the polygon fill algorithm overcomes the drawbacks and fills the interior points of the polygon with the specified colour. The scan line y passes through a connecting vertex, where both the connecting edges are lies in the opposite direction. So at this point 4, the counting will be considered once.

Using the concept of a sorted edge table, the performance of the Polygon fill algorithm can increase. The sorted edge table contains all the information about a scan line required to perform the fill activity including the polygon boundary. During the traversing of the polygon edges, clockwise

or anticlockwise direction, the bucket sort algorithm is used to store the y coordinate.

5.3.1 Global Edge Table

To compute the scan line intersection points made with the polygon edges two data structures namely - Global Edge table (GET) and Active Edge Table (AET) are used. The GET consists of an array of sizes Y_{min} to Y_{max}. The first entry of the array is pointed to the node with Y_{min} as shown in figure 4.

- GET Contains all edges, sorted by Y coordinates in ascending order, i.e. Y_{min} to Y_{max}.
- An array of a bucket is maintained to keep the track of each scan line in the polygon.
- Each bucket has to be maintained as a linked list with increasing X coordinates order where X is the X value at the Y_{min} of the scan line.
- Each entry contains Y_{max}, X_{min} (X at Y_{min}), and 1/m. Where m is the slope of the polygon edge.

Y_{max}	X_{min}(at Y_{min})	1/m

5.3.2 Active Edge Table

Active Edge Table (AET) gives the final intersecting points made by the scan line with the polygon edges with help of information maintained in the Global Edge Table. Active Edge Table starts with a NULL entry. The linked node with the Y_{min} will be added to the AET from the GET.

- Y=minimum, Y value in global edge table (GET) index of the first non-empty bucket.
- Set the active edge table (AET) to be empty
- Repeat until AET and GET are empty.
  - Move edges in GET with Y_{min} = Yinto AET for new edges.
  - Sort AET on X

- Use a list of X coordinates to fill spans
- Remove entries in AET with Y_{max} = Y
- Add 1 to y coordinate to go to the nextscan line
- Update X values of remaining entries in AET
- For NULL entry in GET, set X at Y_{min} = X + 1/m, and Y_{min} and 1/m will remain the same.

Example1: Let us consider a polygon ABCD with some specified coordinates A(2, 1), B(2, 4), C(4, 6), and D(4, 3) as shown in figure 3. Compute the coordinates of the intersecting points made by the scan lines with the polygon edges.

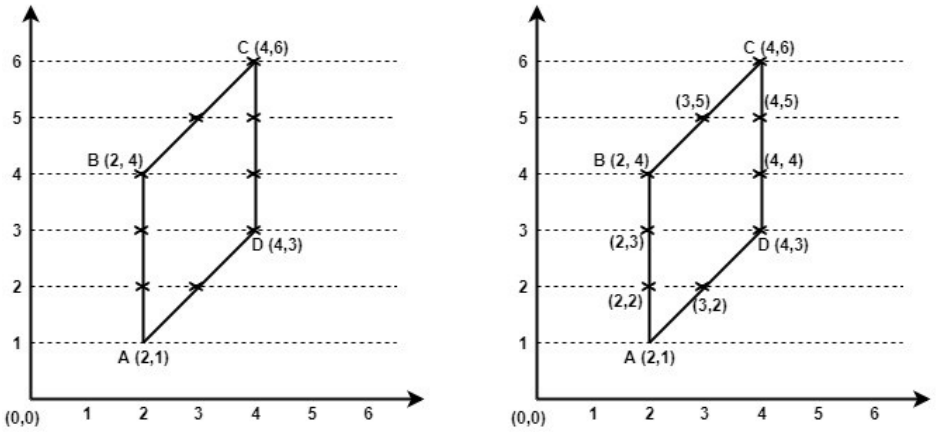


Figure 1: Illustrates the intersecting points made by scan lines with the polygon edges

Solution:The global edge table for the polygon will be-

Sort the vertices in ascending order of Y coordinate i.e. Y_{min} to Y_{max},

I.e. A (2, 1), D(4, 3), B(2, 4), C(4, 6)

Now for the Global edge table entry,

1. The vertex A(2, 1) is connected with D(4, 3) and B(2, 4), So the list entry will for the vertices A(2, 1), D(4, 3)

X1=2 and x2=4

Y1=1 and y2=3

So, Y_{max} = 3 and X_{min} where Y is minimum = 2 and

$$m = \frac{y2-y1}{x2-x1} = \frac{3-1}{4-2} = \frac{2}{2}$$

3	2	1

2. Among the vertex A(2, 1) and B(2, 4)

   Among the vertex A, B

X1=2 and x2=2

Y1=1 andy2=4

So, Y_{max} = 4 and X_{min} where Y is minimum = 2 and

$$m = \frac{y2-y1}{x2-x1} = \frac{4-1}{2-2} = \frac{3}{0}$$

4	2	0

3. Among the vertex D(4, 3), C(4, 6)

X1=4 and x2=4

Y1=3 and y2=6

So, Y_{max} = 6 and X_{min} where Y is minimum = 4 and

$$m = \frac{y2-y1}{x2-x1} = \frac{6-3}{4-4} = \frac{3}{0}$$

6	4	0

4. Among the vertex B(2, 4), C(4, 6)

X1=2 and x2=4

Y1=4 and y2=6

So, Y_{max} = 6 and X_{min} where Y is minimum = 2 and

$$m = \frac{y2-y1}{x2-x1} = \frac{6-4}{4-2} = \frac{2}{2}$$

6	2	1

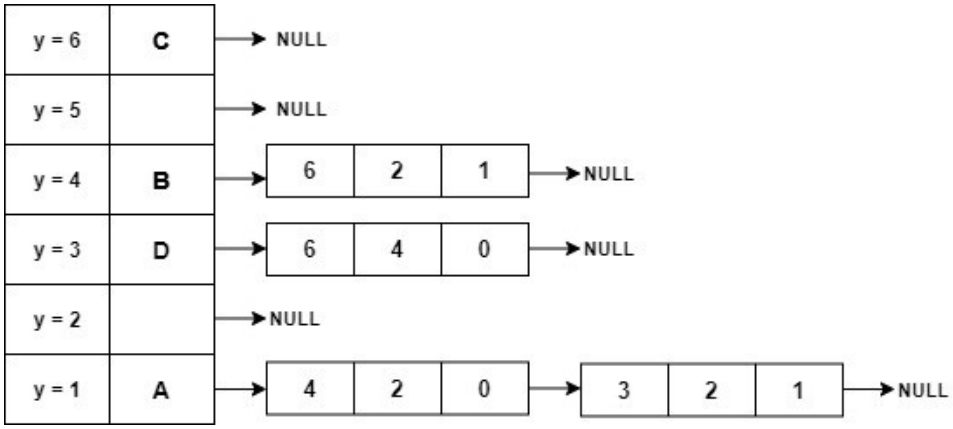
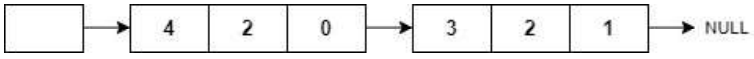


Figure 1: Global Edge Table

Active Edge Table (AET)

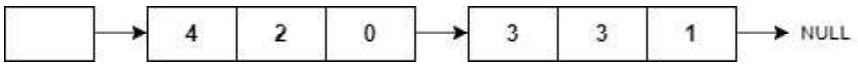
Active Edge Table always starts with a Null entry.

Here the Y_{min} = 1, so the iteration will start from

i = 1 

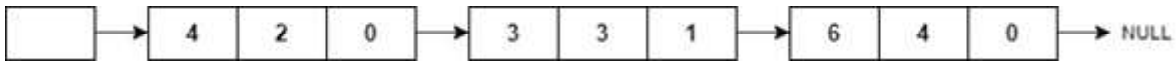
Thus the first intersecting coordinate is (2, 1) and (2, 1)

i = 2, is a NULL entry

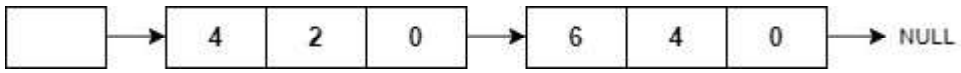


Thus the second intersecting coordinate is (2, 2) and (3, 2)

i = 3, the node connected with D has to be added to the previous node list as shown below.

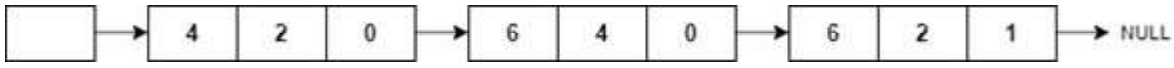


Now the linked list consists of 3 nodes,from which we cannot make pairs of coordinates positions. So the node with Y_{max} equivalent with the y value 3 will be deleted. Finally, the list becomes

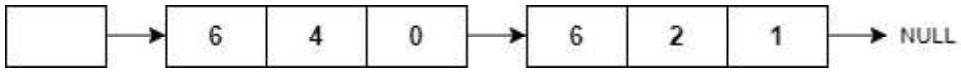


Thus the third intersecting coordinate will be (2, 3) and (4, 3)

i = 4, the node connected with B has to be added to the previous node list as shown below.

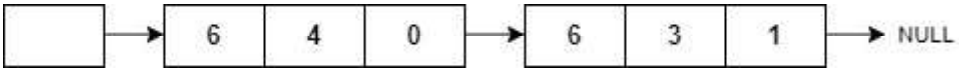


Now the linked list consists of 3 nodes, from which we cannot make pairs of coordinate's positions. So the node with Y_{max} equivalent with the y value 4 will be deleted. Finally, the list becomes



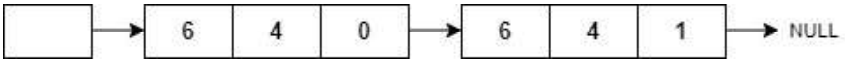
Thus the fourth intersecting coordinate will be (2, 4) and (4, 4)

i = 5, is a NULL entry



Thus the fifth intersecting coordinate is (4, 5) and (3, 5)

i = 6, is a NULL entry



Thus the sixth intersecting coordinate is (4, 6) and (4, 6)

Here the Y_{max} equivalent with the y value 6, so we can delete this two-node, and finally, the list contains the Null node, from where it was started. At the NULL node, stop the further iteration and terminate the processing.

Hence the intersecting coordinates observed during the polygon fill algorithm made by the scan lines with the polygon edges are (2, 1), (2, 1), (2, 2), (3, 2), (2, 3), (4, 3), (2, 4), (4, 4), (4, 5), (3, 5), (4, 6) and (4, 6). The intersecting points are depicted in figure 3.



CHECK YOUR PROGRESS

Q 1: Explain the data structure used by the scan line polygon fill algorithm. Determine the content of the active edge table to fill the polygon with the vertices A(2, 4) B(2, 7) C(4, 9), and D(4, 6).

Q 2: Expect at boundary edges, adjacent pixels are likely to have some characteristics. This property is referred to as ___.

(a) Coherence

(b) Special coherence

(c) Spatial coherence

(d) None of the above

Q 3: The characteristics of pixels on a given scan line change only where a polygon ___ scan line

(a) Intersects

(b) Unions

(c) Differences

(d) All of the above

The coherence property of a scan line can be that the properties of one scene can be linked to the attributes of another section of the scene, allowing the relationship to be exploited to speed up processing.Incremental calculations are frequently used in coherence methods, either along a single scan line or between succeeding scan lines.

5.4 INSIDE-OUTSIDE TEST

In computer graphics, filling the area of an object such as a polygon, circle, ellipse, any bounded region, etc. is a common requirement. This section covers the topic of area filling for standard polygon shapes. Mostly the polygon shapes in geometry are quite simple and easy to identify the interior and exterior points. Common polygon shapes such as triangle, quadrilateral, parallelogram, hexagon, and octagon do not contain any intersecting edges. But in computer graphics, it may have some polygon where edges may intersect and form some irregular patterns as shown in figures 5 and 6. In such cases, it is very difficult to identify the interior or exterior portion of the polygon. Computer graphics packages are used two methods namely the odd-even rule and non-zero winding number rule to identify the interior and exterior regions of a polygon.

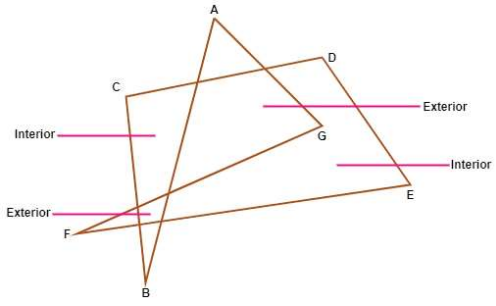


Figure 5 Odd Even rule

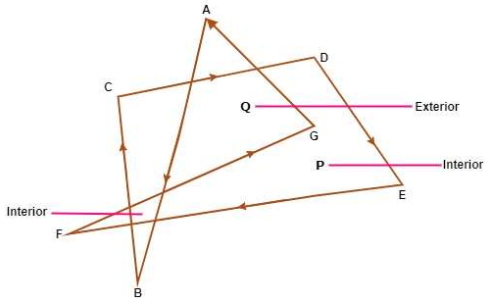


Figure 1 Nonzero Winding number rule

5.4.1 Odd-Even Rule

Odd-even rule or odd parity rule is used to check a region of a polygon, whether it is interior or exterior to the polygon boundary. The Odd-even rule is used to draw a line from a point P from the

polygon towards the outside. The number of polygon edges crossed by the line along its path determines the region status. If the polygon edges crossed by the line are odd, then the point P is considered an interior point, otherwise, P is an exterior point. To obtain an accurate edge count, it must be ensured that the line should not pass through any connecting vertices of polygon edges. Figure 5 depicts the odd-even rule used to identify the interior and exterior points for the given polygon.

5.4.2 Non-Zero Winding Number Rule

The non-zero winding number rule is another approach for defining interior sections of a polygon. This method counts the number of edges winding around a given point, in the scene, in an anticlockwise direction. That counting of the polygon edges is known as the winding number. The points with non-zero winding numbers will be defined as the interior points. For the application of the non-zero winding number rule to an object, first set the winding number count to zero (0), and draw a line from any point on the object extent to the outside coordinates. To get a better result, the line should not pass through a connecting vertex of polygon edges. It has to count the number of polygon edges cross by the line along its line path from a point P to a distant point beyond the polygon boundary in any direction. For every edge with anticlockwise direction or traverse right to left are intersected by the line, add 1 to the counter, and for every edge with clockwise direction intersected by the line add - 1 to the counter. After calculation of the number of edges intersected by the line in the clockwise or anticlockwise direction, the final counter value, i.e. the winding number gives us the relative position of the point P. P is defined as an interior point for a non-zero winding number. Otherwise, P will be considered as an exterior point. Figure 6, depicts interior and exterior points on a polygon defined using the winding number rule.

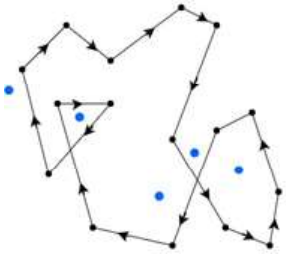
For standard polygon shape, both the odd-even rule and winding number rule results in the same, but for some complex or irregular polygon containing a self-intersecting set of edges, may show the different result as shown in figure 5 and 6.

The non-zero winding number rule is more versatile than the odd-even rule, so some graphics system uses the non-zero winding number rule to implement area filling.

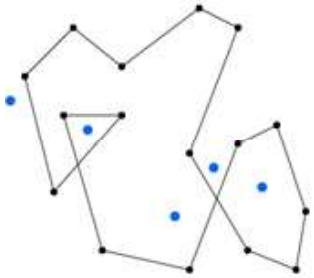


EXERCISE

Question1: Test the dot (highlighted positions) positions, whether they are interior or exterior to the given polygon using non zero winding number rule.



Question2: Test the dot (highlighted positions) positions are interior or exterior to the polygon using odd even rule.



5.5 BOUNDARY FILL ALGORITHM

Another algorithm used to fill the area of an object is the boundary fill algorithm. In this approach, area filling starts with a point inside a region and paints from the interior towards the outside boundary. If the object border is defined by a single colour, the fill algorithm will proceed pixel by pixel until it reaches the boundary colour. In computer graphics painting

packages where the interior points may be detected easily, the boundary fill approach is commonly utilized. An artist can draw a figure's outline and choose fill colour or pattern as well as an interior point. The system may then paint the figures inside.

Input parameters required for the boundary fill algorithm are the coordinate of an interior point (x, y), a fill colour, and a boundary colour. Starting with the inside point coordinate, the system will examine its surrounding pixels and finish filling the area if they match the boundary colour. If not, the pixels will be painted with specified colour and recursively the neighbour will be tested. This process will continue until all pixels up to the boundary are not coloured.

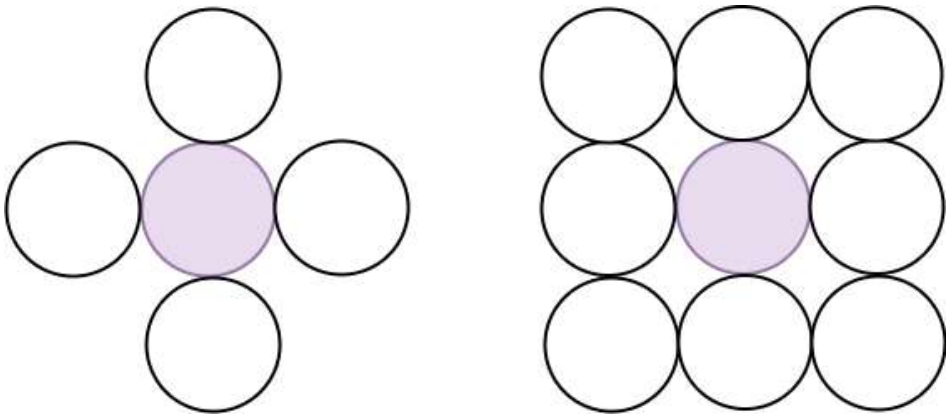


Figure 1: 4 - Connected and 8 - connected Boundary fill algorithm

Figure 7 depicts the two most popular and efficient boundary fill algorithms. In the first method, the system tests the four neighbours, above, below, left, and right of the specified interior pixel point. This process is known as the 4-connected area filling approach. The other method checks the four diagonal pixels including the left, right, above, and below of the current pixel as shown in figure 7, i.e. in this method total of 8 pixels has to be tested by the system to proceed outwards towards the object boundary. If the colours of these neighbour pixels are not matched with the specified boundary colour, the process will continue until it encounters the boundary colour. The fill method using this approach is known as the 8-connected. The 8-connected approach is used in comparatively complex figures and performs better results in comparison to the 4-connected approach. The

recursive boundary fill algorithm has some drawbacks such as if one or more of the interior pixels are coloured with the boundary specified colour, then the boundary fill algorithm will fail to paint the interior pixels. In this approach colour of the neighbour pixels of the current pixel is tested recursively whether they matched the boundary colour or not. So if it found any pixel with the specified boundary colour, the remaining pixels will be unfilled. To avoid this situation, the colour of the interior pixel may be reset before applying the boundary fill algorithm.

5.5.1 4- Connected Area Fill Algorithm

Step 1: set the interior pixel (seed point)coordinate (x, y), its default colour (DC), fill colour (FC), Boundary colour (BC).

Step 2: Specify the polygon boundary value.

Step 3: Test the seed colour, if it is in default colour, then repeat the following steps and till the system encountered boundary specified colour.

If getpixel(x, y) = DC, then repeat step 4 and 5

Step 4: Change the DC value of the seed point with the FC.

SetPixel(x, y, FC)

Step 5: For 4 neighbour pixel (above, below, lef, and right) of the seed point, recursively call the function -

```
FloodFill(x, y+1, FC, DC)     //above the seed point
FloodFill(x, y-1, FC, DC)     //below the seed point
FloodFill(x-1, y, FC, DC)     //left of the seed point
FloodFill(x+1, y, FC, DC)     //right of the seed point
```

Step 6: Stop

5.5.2 8-Connected Area Fill Algorithm

Step 1: set the interior pixel (seed point)coordinate (x, y), its default colour (DC), fill colour (FC), Boundary colour (BC).

Step 2: Specify the polygon boundary value.

Step 3: Test the seed colour, if it is in default colour, then repeat the

following steps and till the system encountered boundary specified colour.

If getpixel(x, y) = DC, then repeat step 4 and 5

Step 4: Change the DC value of the seed point with the FC.

SetPixel(x, y, FC)

Step 5: For 4 neighbour pixel (above, below, left, and right) of the seed point, recursively call the function -

```
FloodFill(x, y+1, FC, DC)    //above the seed point
FloodFill(x, y-1, FC, DC)    //below the seed point
FloodFill(x-1, y, FC, DC)    //left of the seed point
FloodFill(x+1, y, FC, DC)    //right of the seed point
FloodFill(x+1, y+1, FC, DC)  //right top of the seed point
FloodFill(x-1, y+1, FC, DC)  //right left of the seed point
FloodFill(x-1, y-1, FC, DC)  //bottomleft of the seed point
FloodFill(x+1, y-1, FC, DC)  //bottomright of the seed point
```

Step 6: Stop

The 4-connected and 8-connected area fill approaches needs to maintain a stack so as to store the neighbour pixels' information. To overcome this situation another approach has come into the picture. The horizontal pixels across a scan line are filled using this method. The stack initial for each scan line has to maintain in the scan line approach, instead of storing the information regarding 4 or 8 neighbour pixels of the current position. Starting at the initial interior point, fill the continuous span of pixels on the scan line with this approach. Thus the stack starting positions for the next scan line will be located next and the process will continue up to the completion of the area fill for the specified object.



CHECK YOUR PROGRESS

Q 4: In.....algorithm edges can be specified in different colour.

A. flood fill

B. boundary fill

C. DDA fill

D. BSA fill

Q 5: Boundary fill algorithm uses .....call to the fill function.

A. recursive

B. static

C. one

D. none of these

Q 6: Which statement is true about the 4 connected regions?

A. it is faster than 8 connected region

B. sometimes it leaves region unfilled.

C. both a and b

D. neither a or b

Q 7: Active edge table is used in following algorithm

A. boundary fill

B. region fill

C. scan line

D. edge fill

5.6. FLOOD FILL ALGORITHM

The flood fill algorithm is used for area fill when the object boundary contains more than one colour. In this approach, the interior points can paint by replacing a specified colour with the required fill colour instead of searching for the boundary colour. The flood fill algorithm is not concerned about the object boundary colour. Here the process starts from an interior point say (x, y), and reassigns all pixel values that are currently set to a given interior colour with the desired fill colour. In figure 8, the object boundary has four different colours in its boundary such as blue, yellow, orange, and purple. The interior points have a default colour of white. Now, assume that the interior points have to fill with the target colour red. Without testing the boundary colour, the method will start its area filled with the target colour from any interior pixel (x, y). If the colour of the specified

coordinate is white, i.e. the default colour, then it will be replaced by the target colour. To fill the other remaining interior points, the 4 - connected or 8 - connected approach can be utilized. If the interior pixels have more than one colour, then all pixel colours have to reassign to a uniform or solid colour before applying the flood fill algorithm. The interior pixels are coloured using the 4 - connected or 8 - connected area fill approach until all points have been repainted. The 8-connected approach helps to paint the diagonal points (x-1, y-1), (x-1, y+1), (x+1, y+1), and (x+1, y-1) of the current pixel (x,y).

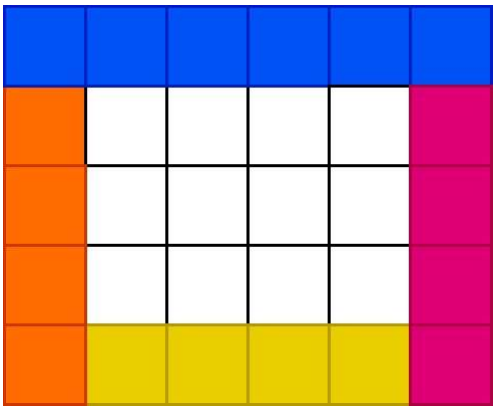


Figure 2: Area covered with a multi-coloured boundary

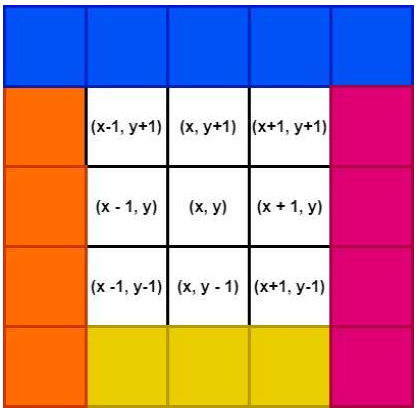


Figure 2Neighbourcoordinates positions for the current pixel (x, y)



5.7 LET US SUM UP

- The polygon fill algorithm locates the intersecting points of the scan lines that make with the polygon edges.
- If the y coordinate of two consecutive endpoints is monotonically decreased or increased, then the middle vertex needs to be counted as a single intersection point for any scan line passing through it.
- In the connecting point where both the polygon edges are lies in the same direction, the count will be considered two times consecutively.
- Using the concept of a sorted edge table, the performance of the Polygon fill algorithm can increase.
- Active Edge Table (AET) gives the final intersecting points made by the scan line with the polygon edges.

- The coherence property of a scan line can be that the properties of one scene can be linked to the attributes of another section of the scene, allowing the relationship to be exploited to speed up processing.
- In the odd-even rule, if the polygon edges cross by the line are odd, then the point P is considered an interior point, otherwise, the P is an exterior point.
- The points with non-zero winding numbers will be defined as the interior points.
- The non-zero winding number rule is more versatile than the odd-even rule.
- In the 4 - connected approach, the system tests the four neighbours, above, below, left, and right of the specified interior pixel point.
- In the 4 - connected approach and the 8 - connected approach colour of the neighbour pixels of the current pixel is tested recursively whether they matched the boundary colour or not.
- In the 4-connected and 8-connected area fill approach, it needs to maintain a stack, storing neighbour pixels' information.
- The flood fill algorithm is used for area fill when the object boundary contains more than one colour.
- The flood fill algorithm is not concerned about the object boundary colour.
- If the interior pixels have more than one colour, then all pixel colours have to reassign to a uniform or solid colour before applying the flood fill algorithm.

5.8 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: 2. A 3. A 4. A 5. A 6. C 7. C

5.9 MODEL QUESTIONS

Short Questions (Answer each question in around 150 words).

Q 1: Choose the correct options from the followings:

i. Scan-line coherence property is used in

A. line drawing algorithm
B. scan line polygon filling algorithm
C. flood fill polygon filling algorithm
D. none of above

ii. In the scan line algorithm the intersection points are

A. sorted in ascending order
B. sorted in descending order
C. not sorted
D. deleted

iii. _____ is a method for testing a pixel inside of a polygon.

A. even-odd method
B. winding number method
C. A and B
D. None of these

iv. _____ is a basic approach used to fill the polygon.

A. seed fill
B. scan fill
C. A and B
D. None of these

v. The seed fill algorithm for filling polygon is classified as _____ fill algorithm and _____ fill algorithm.

A. flood, boundary
B. even, odd
C. edge, flood
D. boundary, scan

vi. Polygon filling algorithms those fill interior-defined regions are called _____ algorithms.

A. flood fill
B. boundary fill
C. scan line
D. edge fill

vii. Polygon filling algorithms those fill boundary-defined regions are called _____ algorithms.

A. flood fill
B. boundary fill
C. edge line
D. A and B

viii. In a boundary fill algorithm for filling polygon, boundary defined regions may be either _____ connected or_____ connected.

A. 2,4
B. 4,8
C. 8,16
D. 8,6

ix. The getpixel function gives the _____ of the specified pixel.

A. intensity
B. colour
C. Size
D. Shape

x. The putpixel function draws the pixel specified_____.

A. intensity
B. colour
C. Size
D. Shape

xi. Seed fill algorithm for filling polygon is _____ algorithm.

A. recursive
B. non-recursive
C. A and B
D. None of these

xii. Scan line algorithm for filling polygon is the _____ algorithm.

A. recursive
B. non-recursive
C. A and B
D. None of these

Q 2: Answer the following questions:

i. What is a polygon filling algorithm?

ii. Explain the working principle of the scan line polygon fill algorithm.
iii. Explain the concept of the boundary fill algorithm.
iv. Write the differences between the 4-connected and 8-connected approaches used in the boundary algorithm.
v. Discuss the advantages and disadvantages of the boundary fill algorithm.
vi. Discuss the advantages of the flood fill algorithm.
vii. Explain the advantages of the scan line polygon fill algorithm.
viii. Write the steps involved in the 4 - Connected and 8 - Connected approaches used in the area fill algorithm.

*** ***** ***

UNIT 6: TRANSFORMATIONS

UNIT STRUCTURE

6.1 Learning Objectives

6.2 Introduction

6.3 Basic concept of transformation

6.4 Types of transformation

6.4.1 Geometric transformation

6.4.1.1 Translation

6.4.1.2 Rotation

6.4.1.3 Scaling

6.4.1.4 Shearing

6.4.2 Coordinate transformation

6.5 Homogenous Coordinate

6.7 Let Us Sum Up

6.8 Further Reading

6.9 Answer to Check Your Progress

6.10 Model Questions

6.1 LEARNING OBJECTIVES

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

- Basic concept of transformation.
- Types of Transformation
- Geometric transformation

6.2 INTRODUCTION

Transformation means moving an object, rotating, resizing an object from one place to another. The need for transformation arises when several object independently defined in its coordinate need to represent in a master coordinate system. It is required to reposition the images on the screen and change their size or orientation.

6.3 BASIC CONCEPT OF TRANSFORMATION

Transformation is any change in the position, shape,or size of an object. There are two types of transformations - geometric transformation, and coordinate transformation. Geometric transformation in which change in the object size, shape, and position takes place,keeping the coordinate system constant. But in coordinate transformation, the object is held constant while the coordinate system is transformed.

6.4 TYPES OF TRANSFORMATION

6.4.1 Geometric Transformation:

Here the objects movesto keep the coordinate system fixed at one place.

TRANSLATION

The translation operation shifts the object from one position to other position. To translate a point from a coordinate position (x,y) to another point (x1,y1),we add algebraically the translation distances Tx and Ty to the original coordinates.

X1=x+Tx

Y1=y+Ty

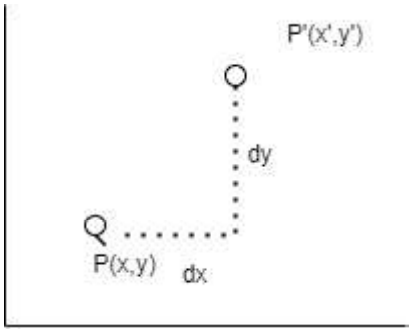
Tx and Ty are called shift vector.

To translate a point from one position to another, matrix representation is:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

Where x' and y' are coordinates of new point,dx and dy are translation distance along x and y axis.

It can be written as P'=P+T where P' is the new position of point P, P is the old position and T is the translation distance.



Question 1. Given a triangle A(2,2),B(10,2) and C(5,5). Translate the triangle with translation distance dx=5,dy=6.

Solution: Given A=(2,2), B(10,2) ,C=(5,5) and translation distance dx=5,dy=6.

We know in order to get new coordinates we have P'=P+t

Now the new coordinates for A ,B,C can be written as

A'=A+T

B'=B+T

C'=C+T

Where A',B',C' are new coordinates after translation and A,B,C are original coordinates and

T represents translation distance along the x and y-axis.

$$A' = \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

$$B' = \begin{bmatrix} 10 \\ 2 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 15 \\ 8 \end{bmatrix}$$

$$C' = \begin{bmatrix} 5 \\ 5 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \end{bmatrix}$$

Thus the new coordinates after translation is A'(7,8),B'(15,8),C'(10,11)

ROTATION

Rotation is a process of changing the angle of the object. It can be clockwise or anticlockwise.

Let coordinate of P in the polar form

X=rcos φ

Y=rsin φ

Where φ is the angle between x-axis and point P, r is the distance from origin to P.

After rotation the new coordinates of the point P is represented by P'(x',y')

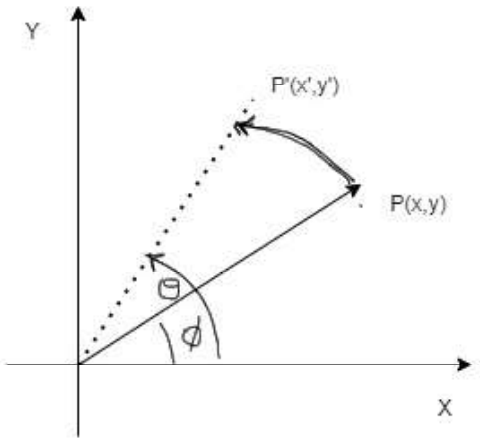
X'=rcos($\varphi + \varphi$)

Y'=rsin($\varphi + \varphi$)

So finally, we can say that

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

i.e P'=R.P



Question 2. Given a triangle whose coordinates are A (2,2),B(8,2) and C(5,5). Rotate the triangle by 90degree.

Solution. We know rotation matrix is given by following.

$$R = \begin{bmatrix} \cos 90 & -\sin 90 \\ \sin 90 & \cos 90 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

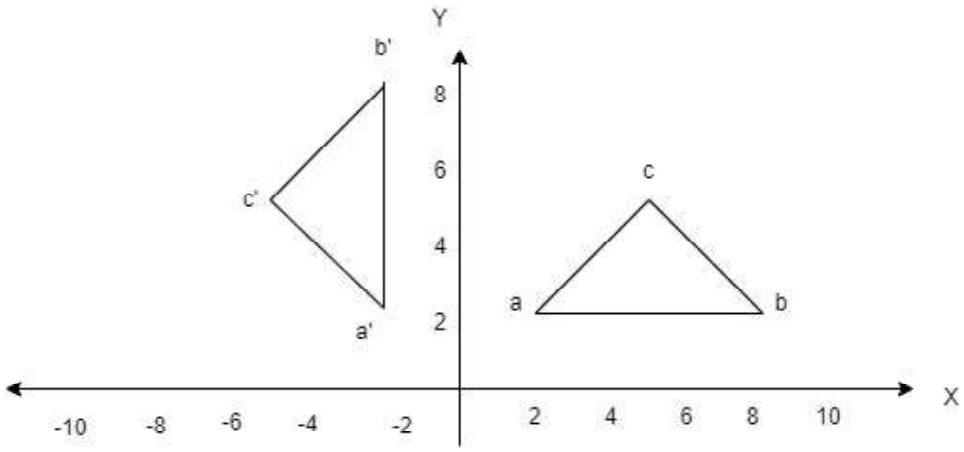
Therefore to get the coordinates of triangle after rotation, we have to multiply rotation matrix with the A,B,C coordinates.

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

$$B' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 8 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 8 \end{bmatrix}$$

$$C' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} -5 \\ 5 \end{bmatrix}$$

Thus the new coordinates A',B',C' after rotation is A'(-2,2),B'(-2,8) and C'(-5,5)



The figure showsthe original triangle which is on the right side and the triangle after rotation is on the left side with coordinates A'(-2,2), B(-2,8), and C'(-5,5).



CHECK YOUR PROGRESS

Q 1: Given a circle C with radius 10 and center coordinates (1, 4). Apply the translation with distance 5 towards X axis and 1 towards Y axis. Obtain the new coordinates of C without changing its radius.

Q 2: Given a triangle with corner coordinates (0, 0), (1, 0) and (1, 1). Rotate the triangle by 90 degree anticlockwise direction and find out the new coordinates.

SCALING

Scaling is used to alter or change the size of an object based on scaling factor. There are two types of scaling factor:

Sx- Scaling which is in x-direction.

Sy-Scaling which is in y-direction

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

$$P' = S.P$$

If scaling factor(SF)=1, the object size will not change.

If SF>1, the object gets expands, and if SF<1, the object size gets reduced.

The above matrix can be represented in 3*3 matrix as written below

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

QUESTION 3. Scale the polygon with coordinates A(2,5),B(7,10) and C(10,2) by two units in x-direction and two units in the Y direction.

Solution: Here given A=(2,5), B=(7,10),(10,2) and Sx=Sy=2

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Putting the value of Sx,Sy and A,B,C in the above,we have

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 7 & 10 \\ 5 & 10 & 2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 14 & 20 \\ 10 & 20 & 4 \\ 1 & 1 & 1 \end{bmatrix}$$

Scaling matrix

3 points of polygon in matrix form

Since we have three points in the polygon A(x1,y1),B(x2,y2) and C(x3,y3) ,so we have written these three points in the above matrix form with scaling matrix.

SHEARING

Shearing transformation makes the object to slant either in the X-direction or in Y-direction. Two types of shearing transformation are there-X -shear and Y-shear.X-shear maintains the y coordinates and changes take place in x values which causes the vertical line to tilt left or right. The Y-shear preserves the x-coordinate but changes take place in y coordinates.

The new coordinates after applying shearing is:

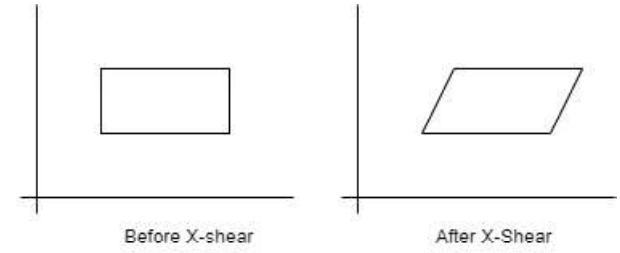
$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ shx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Shearing matrix in X-axis

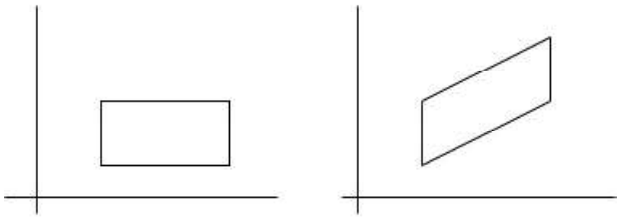
$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Shy & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Shearing matrix in Y-axis

After applying shearing, the effects are shown below:



Before X-shear After X-Shear



Before Y-shear After Y-shear



CHECK YOUR PROGRESS

Q 3: Given a square object with coordinate points A(0, 3), B(3, 3), C(3, 0), D(0, 0). Apply the scaling parameter 2 towards X axis and 3 towards Y axis and obtain the new coordinates of the object.

Q 4: Given a triangle with points (1, 1), (0, 0) and (1, 0). Apply shear parameter 2 on X axis and 2 on Y axis and find out the new coordinates of the object.

6.4.2 Coordinate Transformation

We can apply the effects of translation, scaling and rotation by moving the observer who views the object and by keeping the object stationary. This type of transformation is called coordinate transformation.

6.5 HOMOGENOUS COORDINATE

A 2-dimensional coordinate can be represented in 3 dimensions using homogenous coordinates. The two-dimensional transformation can be represented as 3*3 matrices so that usage of the operations becomes relatively easy. Generally, this concept of homogenous is used in design and construction applications. This conversion is done in translation, scaling, shearing, and rotation to fit the picture into the proper position.

Consider coordinates (x,y) in 2-D. This can be represented in 3 D as (x_h, y_h, h) where

X_h=x. h and Y_h=y. h.

In order to convert from 3-D to 2-D

$X=X_h/h$

$Y=Y_h/h$

Therefore matrix representation of translation in homogenous coordinates is given by:

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 1 & 0 & ty \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Rotation matrix in homogenous is given by:

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

The scaling matrix in homogenous coordinates is given by:

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Y \end{bmatrix}$$

The Shearing matrix in homogenous coordinates is given by

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$



CHECK YOUR PROGRESS

Q 5: Convert coordinates (2,3) into homogenous coordinates if h=2.

Q 6: Convert homogenous coordinates (8,4,2) into 2D.



6.6 LET US SUM UP

- Ability to simulate the manipulation of objects in space is transformation.
- Geometric transformation is in which the object itself is transformed relative to a stationary background.
- Coordinate transformation in which the object is held stationary while coordinate system is transformed relative to the object.
- In translation, an object is displaced a given distance and direction from its original position.

- In rotation, the object is rotated theta degree about the origin.
- Scaling is the process of expanding or compressing the dimensions of an object.
- In a homogenous coordinate system, two-dimension coordinate positions are transformed into three-dimensioncoordinates.



6.7 FURTHER READINGS

1) Computer Graphics y Zhigang Xiang and Roy A Plastock
2) Computer Graphics By Hearn and Baker



6.8 ANSWER TO CHECK YOUR PROGRESS

Ans to Q No 1: New center coordinates of C = (6, 5).

Ans to Q No 1: New coordinates of the triangle after rotation = A (0, 0), B(0, 1), C(-1, 1).

Ans to Q No 1: New coordinates after scaling =A(0,9),B(6,9),C(6,0),D(0,0)

Ans to Q No 1: New coordinates of the triangle after shearing in Y axis = A (1, 3), B(0, 0), C(1, 2).

New coordinates of the triangle after shearing in X axis = A (3, 1), B(0, 0), C(1, 0).

Ans to Q No 1: Homogenous coordinates of (2,3) is (4,6,2)

Ans to Q No 1: x=8/2=4,y=4/2=2 So the original coordinates in 2-D is (4,2).



6.9 MODEL QUESTIONS

Q 1: Find the matrix that represents rotation of an object by 30 degree about the origin.

Q 2: What are the new coordinates of the point P(2,-4) after the rotation?

Q 3: Write the general form of the matrix for rotation about a point P(h,k).

Q 4: Perform a 45 degree rotation of a triangle A(0,0),B(1,1) ,C(5,2) (a) about a origin (b)about P(-1,-1)

Q 5: Write the general form of a scaling matrix with respect to a fixed point P(h,k)

Q 6: Magnify the triangle with vertices A(0,0),B(1,1),C(5,2) to twice its size while keeping C(5,2) fixed.

Q 7: Prove that 2D Scaling transformations are commutative i.e, S1 S2=S2 S1.

Q 8: Prove that 2D rotations about the origin are commutative i.e. R1 R2=R2 R1.

Q 9: Rotate a line CD whose endpoints are (3, 4) and (12, 15) about origin through a 45° anticlockwise direction.

Q 10: Rotate line AB whose endpoints are A (2, 5) and B (6, 12) about origin through a 30° clockwise direction.

*** ***** ***

UNIT 7: 2D VIEWING

UNIT STRUCTURE

7.1 Introduction

7.2 Unit Objectives

7.3 Basic concept of Viewing pipelining

- 7.3.1 Viewing pipeline
- 7.3.2 Clipping
- 7.3.3 Clipping Algorithm
- 7.3.4 Application of clipping

7.4 Viewing coordinate reference frame

7.5 Window to viewport coordinate transformation

7.6 Viewing function

- 7.6.1 PHIGS function
- 7.6.2 OPENGL function

7.7 Let Us Sum Up

7.8 Further Reading

7.9 Answer to Check Your Progress

7.10 Model Questions

7.1 LEARNING OBJECTIVES

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

- Able to understand the procedures for displaying views a two-dimensional picture on an output device.
- Able to know about the clipping procedure which isused to identify those portions of a picture that are either inside or outside a specified region of space.

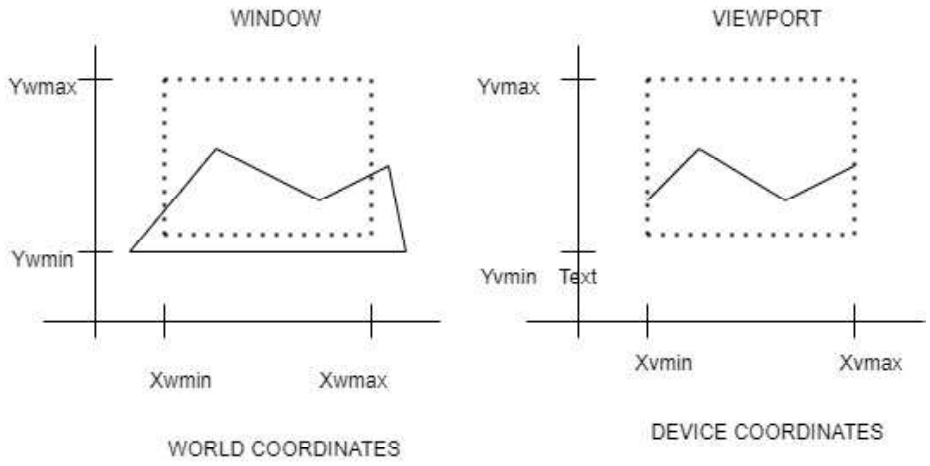
7.2 INTRODUCTION

A graphics package allows a user to specify where part of a defined picture is to be displayed and where that part is to be displayed on the

display device using clipping. Any conventional cartesian coordinate system, referred to as the world coordinate reference frame can be used to define the picture. For a 2D picture, a view is selected by specifying a subarea of the total sub-area.The transformation from world to device coordinates involves translation, rotation, and scaling operations as well as procedures for deleting those parts of the picture that are outside the limits of a selected display area.

7.3 BASIC CONCEPT OF VIEWING PIPELINE

7.3.1 Viewing Pipeline



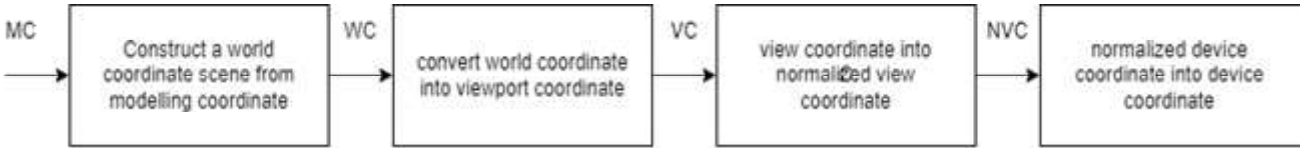
An image representing a view often becomes part of a larger image, like a photo on an album page, which model a computer monitor's display area. Since the monitor sizes differ from system to system, we want to introduce a device-independent tool to describe the display area. This tool is called a normalized device coordinate system (NDCS).

A world coordinate area selected for display is called a window. An area on a display device to which a window is mapped is called a viewport. In simple terms, what is to be display is called window and where it is to be displayed is called a viewport.

Mapping of a part of a world coordinate scene to device coordinate is referred to as Viewing transformation.Sometimes,the 2-D viewing transformation is simply referred to as the window-to-viewport

transformationto the windowing transformation.

In graphics packages, generally, the window we select is the standard rectangular shape. For example, if our window is not a rectangle shape then we have to follow some steps. So the steps are

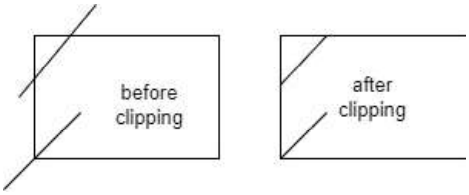


Viewing Pipeline

- Construct a world coordinate screen from modeling co-ordinate.
- Convert this world coordinate into our viewport co-ordinate.
- Once the viewport co-ordinate obtain,transform the description of the world coordinate into the viewport. It means view coordinates into normalized view coordinates.
- This normalized view coordinate is transformed into a device coordinate.

7.3.2 Clipping

The portion which lies outside the region of the window, in computer graphics, is called the clipped part and the process of displaying the image inside the region of the window is called clipping.



The clipping algorithm can be applied in both world coordinate and normalized device coordinates

In world coordinate system, only the contents of the window interior are mapped to device coordinates, world coordinate clipping removes those primitives outside the window from further consideration.

In normalized device coordinate system, mapping is fully done from world to the normalized device. Clipping will reduce calculations by allowing concatenation of viewing and geometric transformation matrices.

7.3.3 Clipping Algorithms

Algorithms have been developed for clipping for the following primitive types:

- Point Clipping.
- Line Clipping (straight-line segments)
- Area Clipping (polygons)
- Curve Clipping
- Text Clipping.

7.3.4 Application of Clipping

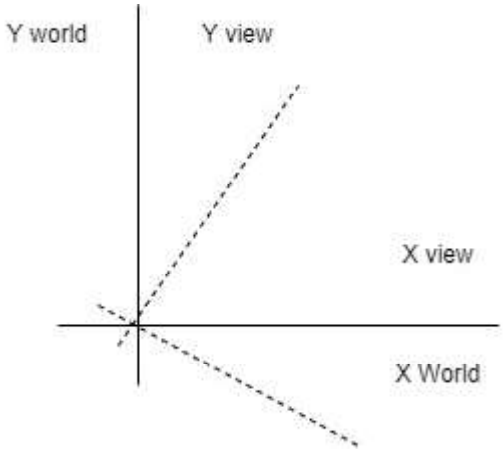
- Extracting part of defined scene for viewing.
- Identifying visible surfaces in three-dimensional views.
- Anti-aliasing line segments or object boundaries.
- Creating objects using solid-modeling procedures.
- Displaying a multiwindow environment.
- Drawing and painting operations that allow parts of picture to be selected for copying, moving,
- Erasing, or duplication.

7.4 VIEWING COORDINATE REFERENCE FRAME

We will have the matrix for converting world coordinate positions to viewing coordinates as a two-step composite transformation

First, the viewing origin is translated to the world origin, then we rotate to align the two coordinate reference frames.The composite 2D transformation to convert world coordinates to viewing coordinatesis where T is the translation matrix that takes the viewing origin point P to the world

origin, and R is therotation matrix that aligns the axes of the two reference frames.



Viewing coordinate reference frame

7.5 WINDOW TO VIEWPORT COORDINATE TRANSFORMATION

A window is specified by four coordinates: wxmin, wxmax, wymin, wymax. Similarly a viewport is described by four coordinates: vxmin, vxmax, vymin, vymax.The objective of window-to-viewport mapping is to convert the world coordinates(wx,wy) of any point to its corresponding normalized device coordinates(vx,vy). In order to maintain relative changes of the point in the viewport as in the window, we require:

$$\frac{wx_{min} - wx_{max}}{wx_{max} - wx_{min}} = \frac{vx - vx_{min}}{vx_{max} - vx_{min}}$$

and

$$\frac{wy - wy_{min}}{wy_{max} - wy_{min}} = \frac{vy - vy_{min}}{vy_{max} - vy_{min}}$$

From the above two equations we have:

$$vx = \frac{vx_{max} - vx_{min}}{wx_{max} - wx_{min}} [wx - wx_{min}] + vx_{min}$$

$$vy = \frac{vy_{max} - vy_{min}}{wy_{max} - wy_{min}} [wy - wy_{min}] + vy_{min}$$

Question 1.

Find the workstation transformation that maps the normalized device screen onto a physical device whose x extent is 0 to 199 and y extent is 0 to 639 where the origin is located at the (a) lower left corner.

Solution:

(a)The window parameters are wx_{min}=0 ,wx_{max}=1,wy min=0 and wy_{max}=1.

The viewport parameters are vxmin=0,vx_{max}=199, vy_{min}=0 andvy_{max}=639.

Then sx=199, s y= 639 and

$$W = \begin{bmatrix} 199 & 0 & 0 \\ 0 & 639 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



CHECK YOUR PROGRESS

Q 1: Express window-to-viewport mapping in the form of a composite transformation matrix.

$$S_x = \frac{vx_{max} - vx_{min}}{wx_{max} - wx_{min}} \quad \text{and} \quad \frac{vy_{max} - vy_{min}}{wy_{max} - wy_{min}}$$

Q 2: Show that for a viewing transformation, S_x and S_y if and only if $a_w = a_v$, where a w is the aspect ratio of the window and a v the aspect ratio of the viewport.

7.6 VIEWING FUNCTION

In a PHIGS application program ,following are the viewing function:

- evaluateViewOrientationMatrix (x0,y0,xv,yverror,viewMatrix)
  where x0,y0 are coordinates of viewing originxv,yvis the world coordinate positions for the view up vector.
- valuateview Mapping Matrix (xwmin, xwmax, ywmin, ywmax, xvmin, yvmax, error view Mapping Matrix)
  The window limits in viewing coordinates are chosen with parameters xwmin, xwmax, ywmin, ywmax and the viewport limits are set with the normalized coordinate positions xvmin,xvmax,yvmin, yvmax.
- set View Representation(ws, viewindex, view Matrix, view Mapping Matrix, xclipmin, xclipmax, yclipmin, yclipmax, clipxy)
  where parameter ws designate the output device(workstation) and parameter viewindex sets an integer identifier for this particular This allows us to turnoff clipping if we want to view the scene outside the viewport.

7.6.1 Opengl 2d Viewing Function

- To transform from world coordinate to screen coordinates, the appropriate matrix mode must be chosen
  glMatrixMode (GL_PROJECTION); glLoadIdentity( );
- To define a 2D clipping window, we use OpenGL Utility function:
  gluOrtho2D(xwmin, xwmax, ywmin, ywmax);
  This function also perform normalization (NDC)
- To specify the viewport parameters in OpenGL, we use function
  glViewport(xvmin, yvmin, vpWidth, vpHeight);



7.7 LET US SUM UP

- Four Cartesian co-ordinates systems in computer Graphics.
  1. Modeling co-ordinates
  2. World co-ordinates

3. Normalized device co-ordinates

4. Device co-ordinates

- Modeling Coordinates are also known as local coordinates, where the individual object in a scene is within separate coordinate reference frames. Each object has an origin (0,0)
- World Co-ordinates: World coordinate system describes the relative positions and orientations of every generated object.The scene has an origin (0,0). The object in thescene is placed with reference to the scenes origin. The world co-ordinate scale may be the same as the modeling co-ordinate scale or it may be different.
- Normalized Device Co-ordinates. Output devices have their own coordinates. Co-ordinates values: The x and y axis range from 0 to 1. All the x and y coordinates are floating-point numbers in the range of 0 to 1.This makes the system independent of the various device coordinates. This is handled internally by graphic system without user awareness.
- Device Co-ordinates: Specific co-ordinates used by a device.
- When we interested to display certain portion of the drawing, enlarge the portion, windowing technique is used.
- Technique for not showing the part of the drawing which one is not interested is called clipping
- An area on the device (ex. Screen) onto which the window will be mapped is called viewport.
- Window defines what to be displayed.
- A viewport defines where it is to be displayed.
- Most of the time, windows and viewports are usually rectangles in standard position(i.e aligned with the x and y axes). In some application, others such as general polygon shape and circles are also available



7.8 FURTHER READINGS

1) Schaum's Outline of Computer Graphics 2/E by Zhigang Xiang and Roy A Plastock

2) Computer Graphics By Donald Hearn and M. Pauline Baker

7.9 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1:

$$N = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -wx_{min} & -wy_{min} & 1 \end{bmatrix} \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ vx_{min} & vymin & 1 \end{bmatrix}$$

Ans to Q No 1: if $S_x = S_y$ then

$$\frac{vx_{max} - vx_{min}}{wx_{max} - wx_{min}} = \frac{vy_{max} - vy_{min}}{wy_{max} - wy_{min}}$$

or

$$\frac{wy_{max} - wy_{min}}{wx_{max} - wx_{min}} = \frac{vy_{max} - vy_{min}}{vx_{max} - vx_{min}}$$

7.10 MODEL QUESTIONS

Short Questions (Answer each question in around 150 words).

Q 1: What do you mean by window-to-viewport mapping? Explain.

Q 2: What are the different viewing functions available in PHIGS?

Q 3: What is Clipping? What are different types of clipping available.

Q 4: Define window, viewport.

Q 5: Why window to viewport mapping required?

Q 6: Find the normalization transformation that maps a window whose lower left corner is at (1,1) and upper right corner is at (3,5) onto

(a) A viewport that is the entire normalized device screen

(b) A viewport that has lower left corner at (0,0) and upper right corner (1/2,1/2).

Q 7: Find the normalization transformation which uses a circle of radius five units and center(1,1) as a window and a circle of radius ½ and center (1/2,1/2) as a viewport.

Q 8: Explain modeling coordinates and device coordinates with example.

Q 9: What is the condition to be satisfied for a point to be inside the window?

Q 10: What are viewing functions available in OPENGL?

*** ***** ***