KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Coimbatore-641 021
(For the candidates admitted from 2017 onwards)
DEPARTMENT OF COMPUTER SCIENCE, CA & IT

SUBJECT CODE : 17CSU602B SUBJECT : COMPUTER GRAPHICS
SEMESTER : VI CLASS : III B.Sc. CS L T P C = 4 0 0 4

Course Outcome

This course presents an introduction to computer graphics designed to give the student an overview of fundamental principles. The course will include an overview of common graphics hardware, 2D and 3D transformations and viewing, and basic raster graphics concepts such as scan-conversion, and clipping. Methods for modeling objects as polygonal meshes or smooth surfaces, and as rendering such as hidden-surface removal, shading, illumination, and shadows will be investigated.

Learning outcome

- Have a knowledge and understanding of the structure of an interactive computer graphics system, and the separation of system components.
- Have a knowledge and understanding of geometrical transformations and 3D viewing.
- Be able to create interactive graphics applications.
- Have a knowledge and understanding of techniques for representing 3D geometrical objects.
- Have a knowledge and understanding of the fundamental principles of local and global illumination models.

UNIT-I

Introduction : Basic elements of Computer graphics, Applications of Computer Graphics.

UNIT-II

Graphics Hardware : Architecture of Raster and Random scan display devices, input/output devices.

UNIT-III

Fundamental Techniques in Graphics : Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

UNIT-IV

Geometric Modeling : Representing curves & Surfaces.

UNIT V

Visible Surface determination : Hidden surface elimination. **Surface rendering :** Illumination and shading models. Basic color models and Computer Animation.

TEXT BOOKS:

T1: Donald Hearn, Computer Graphics C Version,Second edition,Pearson Edition,2011.

SUGGESTED READINGS

R1:Foley, J.D., Van Dan, A.,& Feiner Hughes. (1990). Computer Graphics Principles & Practice (2nd ed.). Addison Wesley.

R2:Hearn D. Baker. (2008). Computer Graphics. New Delhi: Prentice Hall of India.

R3: Rogers, D.F. (1997). Procedural Elements for Computer Graphics. McGraw Hill.

R4: Rogers, D.F. Adams. (1989). Mathematical Elements for Computer Graphics(2nd ed.). McGraw Hill.

WEBSITES:

W1: study.com/academy/lesson/what-are-the-seven-elements

W2: www.linkedin.com/pulse/application-computer-graphics-niropam-das

W3: en.wikipedia.org/wiki/Graphics_hardware
W4:en.wikipedia.org/wiki/Hidden_surface_determination

ESE MARKS ALLOCATION

1.	**Section A** 20 x 1 = 20	20
2.	**Section B** 5 x 2 = 10	10
3.	**Section C** **5 x 6= 30** Either ‘A’ or ‘B’ choice	30
	Total	60

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University) (Established
Under Section 3 of UGC Act 1956)

Coimbatore - 641021.

(For the candidates admitted from 2017 onwards)

DEPARTMENT OF COMPUTER SCIENCE, CA & IT

SUBJECT : COMPUTER GRAPHICS

SEMESTER :VI **L T P C**

SUBJECT CODE:17CSU602B CLASS :III B.Sc CS **4 0 0 4**

LECTURE PLAN

STAFF NAME: Dr P.TAMIL SELVAN

S.No	Lecture Duration (Hr)	Topics	Support Materials
		UNIT-I	
1.	1	Introduction	T1:22
2.	1	A Survey of Computer Graphics	T1:22-54
3.	1	Basic elements of Computer graphics-Line,Shape	T1:55-58
4.	1	Basic elements of Computer graphics-Color,Texture	T1:59-61
5.	1	Basic elements of Computer graphics-Value,Space	T1:61-63
6.	1	Applications of Computer Graphics	T1:64-65
7.	1	Applications of Computer Graphics[Cont..]	T1:65-68,W2
8.	1	Recapitulation and Discussion of Important Questions	
		Total No of Hours Planned For Unit – I	**8**
		UNIT-II	
1.	1	Graphics Hardware	**W3**
2.	1	Raster Scan Displays	T1:60
3.	1	Random Scan Displays	T1:61-62,
4.	1	Input devices :Keyboards, Mouse	T1:81
5.	1	Track Ball and space ball, Joysticks, Data Glove,	T1:82-84,

		digitizers	
6.	1	Output devices: Hard Copy Devices: Printers and Plotters	T1:92
7.	1	Recapitulation and Discussion of Important Questions	
		Total No of Hours Planned For Unit – II	**7**
		UNIT-III	
1.	1	Fundamental Techniques in Graphics Raster scan line	T1:104-105
2.	1	Line Drawing Algorithms	T1:106-107
3.	1	DDA Algorithm	T1:107-108
4.	1	Circle Generating Algorithms	T1:117-118
5.	1	Mid Point Circle Algorithm	T1:118—121
6	1	ellipse drawing	T1:122
7	1	thick primitives	T1:122
8	1	Polygon filling	T1:257
9	1	line and polygon clipping algorithms	T1:245-255
10	1	Two Dimensional Geometric Transformations: Basic Transformations	T1:204-208
11	1	2D,3D Viewing transformations	T1:237
12	1	Recapitulation and Discussion of Important Questions	
		Total No of Hours Planned For Unit – III	**12**
		UNIT-IV	
1.	1	Geometric Modeling	T1:427
2.	1	Representing curves	T1:330
3.	1	Surfaces	T1:330-331
4.	1	Recapitulation and Discussion of Important Questions	
		Total No of Hours Planned For Unit – IV	**4**
		UNIT-V	
1.	1	Visible Surface determination	T1:490
2.	1	Hidden surface elimination	W4
3.	1	Surface rendering	T1:514
4.	1	Illumination and shading models	T1:543
5.	1	Basic color models and Computer Animation.	T1:584
6.	1	Recapitulation and Discussion of Important Questions	
7.	1	Discussion of Previous ESE Question Papers	
8.	1	Discussion of Previous ESE Question Papers	
9.	1	Discussion of Previous ESE Question Papers	
		Total No of Hours Planned For Unit – V	**9**

		Total No. of Hours Planned:	**40**

TEXT BOOKS:

T1: Donald Hearn, Computer Graphics C Version,Second edition,Pearson Edition,2011.

SUGGESTED READINGS

R1: Foley, J.D., Van Dan, A.,& Feiner Hughes. (1990). Computer Graphics Principles & Practice (2nd ed.). Addison Wesley.

R2:Hearn D. Baker. (2008). Computer Graphics. New Delhi: Prentice Hall of India.

R3: Rogers, D.F. (1997). Procedural Elements for Computer Graphics. McGraw Hill.

R4: Rogers, D.F. Adams. (1989). Mathematical Elements for Computer Graphics(2^{nd} ed.). McGraw Hill.

WEBSITES:

W1: study.com/academy/lesson/what-are-the-seven-elements

W2: www.linkedin.com/pulse/application-computer-graphics-niropam-das

W3: en.wikipedia.org/wiki/Graphics_hardware
W4:en.wikipedia.org/wiki/Hidden_surface_determination

UNIT-I

SYLLABUS

Introduction : Basic elements of Computer graphics, Applications of Computer Graphics.

INTRODUCTION:

Computer Graphic is the discipline of producing picture or images using a computer which include modeling, creation, manipulation, storage of geometric objects, rendering, converting a scene to an image, the process of transformations, rasterization, shading, illumination, animation of the image, etc.

Computer Graphics has been widely used in graphics presentation, paint systems, computer-aided design (CAD), image processing, simulation, etc.

From the earliest text character images of a non-graphic mainframe computers to the latest photographic quality images of a high resolution personal computers, from vector displays to raster displays, from 2D input, to 3D input and beyond, computer graphics has gone through its short, rapid changing history.

From games to virtual reality, to 3D active desktops, from unobtrusive immersive home environments, to scientific and business, computer graphics technology has touched almost every concern of our life. Before we get into the details, we have a short tour through the history of computer graphics

BASIC ELEMENTS OF COMPUTER GRAPHICS:

There are in total six elements of a design which you need to be aware of: the line, the shape, the color, the texture, the value and the space.

1. The line

The line is usually present in every design, even if it is a solid border of 1px or a dotted one of 5px. Every website has lines, but the minimalistic style which became more popular in the past couple of years tries to erase the lines from the layouts, or at least to decrease the use of them.

The lines can be long, red, straight, thin, blue, dashed, short, black or curved, they are all into the same category. They are most of the time used for delimitation between different sections of a design, or are used to direct a viewer's vision in a specific direction.

The lines can create different effects and visual impact. While a thick, bold line draws attention because of its visual power, the thin lines tend to go the other way. The color has an impact too, dark colors are easier to see and draw more attention than light or pale colors.

And this is not all. The style of a line can also influence the way the user sees it. This style can easily be defined through CSS and can be solid, dotted and dashed among others. The solid lines have a different impact than the dotted ones, because they are more imposing.

The minimalistic style which I've talked about earlier uses either less solid lines or more curved lines, because they give a dynamic and fluid look to a design, which is also the purpose of the style. They indicate energy, keep the user interested and, if combined with illustration, are very powerful to the human eye.

Many years ago solid lines were very popular because they determined the style of the design: rigid, solid and organized. The web changed in the past years and this style is not very popular anymore, especially for designers' portfolios and other pages with a strong need of a personal touch.

The solid lines are used to separate different parts of the website.

2. The shape

The shape, or the form, is the second most used element of a web design. They are actually lines combined in different shapes. The forms are still popular and this is because if there is something that needs to stand out, forms are one of the ways to do it.

There can be circles, squares, rectangles, triangles or any other abstract shape; most of the designs include at least one of these. Minimalistic designs use it a lot, because they are often based on illustrations and drawings.

The old style of designing websites included shapes too, so they remained popular throughout the time and will probably continue being like that.

Like lines, shapes are also associated by the human mind with different movements. For example, circles are associated with movement and nature, while squares are often seen as structured, basic designs. Just like with the lines, the color, style, background or texture of a shape can totally change the viewer's perception.

3. Textures

The textures were not very popular a couple of years ago, but they tend to become more and more used. They replaced (or compete with, if we can call it a competition) the single-colored backgrounds.

Textures can look similar to solid background colors, but if they are analyzed closer, small but effective differences can be noticed.

Texture styles include paper, stone, concrete, brick, fabric and natural elements, among flat or smooth colors. Textures can also be subtle or pronounced and can be used sparingly or liberally. They work with pretty much everything.

Even if they do not seem important, the textures can totally change a website and offer a totally different visual impact.

Color

The color may even be the most important element of a design, because it offers the most powerful visual impact at a single glance. Color is obvious and does not need basic graphic skills to be noticed.

While lines and shapes mean the same thing as in the reality, only at a little more profound level, the color means exactly the same thing as in the nature. Color creates emotions – red is passionate, blue is calm, green is natural.

Even if you don't realize this, colors have a clear effect on your mind.

Studies have been done and a person who lives in a red environment has a higher heartbeat and pulse than a person living in a blue environment. The human brain sees this and influences the rest of the body.

Therefore color theory is very important to know, because not many designers can call themselves experts in this field. Being a master of colors might make the difference between a good design and a stunning one.

I am not saying you have to know all of them, but knowing how hue, saturation, shade, tint, tone or chroma work together is crucial for a graphic designer.

5. Value

I did not specify value above, even if it is closely related to color, because value is more general and represents how dark or light a design is. Value has a lot to do with mood too, only at a more profound level.

Understanding colors will take you close to perfection, but knowing how value works will take you beyond this. Lighter designs offer a different impact and feeling than the dark ones and you need an expert eye to notice differences and decide which one is the best.

6. Space

The space and how it is used is crucially important in design. Lately the "white space" (also called negative space) became used widely because it allows the human eye to read easier.

For whoever is not familiar with the term "white space", it does not mean precisely space filled with white, but every area of the design which is only filled with the background color. You can see several examples below to better understand the concept.

If there is a lot of negative space in your web design, it offers light and an open feeling. The lack of white space will turn your design into an old-fashioned, cluttered one. The space has also a lot to do with how the design is perceived by the human eye.

Even if I said the color is maybe the most important element of a design, the space is definitely present in the top, because it is also very easy to notice by the untrained eye. It can turn a design to your advantage and get the best out of your layout.

APPLICATIONS OF COMPUTER GRAPHICS

Computers have become a powerful tool for the rapid and economical production of pictures. There is virtually no area in which graphical displays cannot be used to some advantage, and so it is not surprising to find the use of computer graphics so widespread. Although early applications in engineering and science had to rely on expensive and cumbersome equipment, advances in computer technology have made interactive computer graphics a practical tool. Today, we find computer graphics used routinely in such diverse areas as science, engineering, medicine, business, industry, government, art, entertainment, advertising, education, and training.

1.1.1 Computer-Aided Design

A major use of computer graphics is in design processes, particularly for engineering and architectural systems, but almost all products are now computer designed. Generally referred to as CAD, computer-aided design methods are now routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft, computers, textiles, and many, many other products.

For some design applications, object are first displayed in a wire frame outline form that shows the overall sham and internal features of objects. Wire frame displays also allow designers to quickly see the effects of interactive adjustments to design shapes.

Software packages for CAD applications typically provide the designer with a multi-window environment. Circuits such as the one shown in below Figure and networks for communications, water supply, or other utilities are constructed with repeated placement of a few graphical shapes. The shapes used in a design represent the different network or circuit components. Standard shapes for electrical, electronic, and logic Circuits are often supplied by the design package. For other applications, a designer can create personalized symbols that are to be used to construct the network or circuit. The system is then designed by successively placing components into the layout, with the graphics package automatically providing the connections between components. This allows the designer t~ quickly try out alternate circuit schematics for minimizing the number of components or the space required for the system.

Fig: A circuit design application using

Animations are often used in CAD applications. Real-time animations using wire frame displays on a video monitor are useful for testing performance of a vehicle or system, When we do not display objects with rendered surfaces, the calculations for each segment of the animation can be performed quickly to produce a smooth real-time motion on the screen. Also, wire frame displays allow the designer to **see** into the interior of the vehicle and to watch the behavior of inner components during motion. Animations in virtual reality environments are used to determine how vehicle operators are affected by certain motions. As the tractor operator manipulates the controls, the Headset presents a stereoscopic view of the front-loader bucket or the backhoe, just as if the operator were in the tractor seat. This allows the designer to explore various positions of the bucket or backhoe that might obstruct the operator's view, which can then be taken into account in the overall hector design.

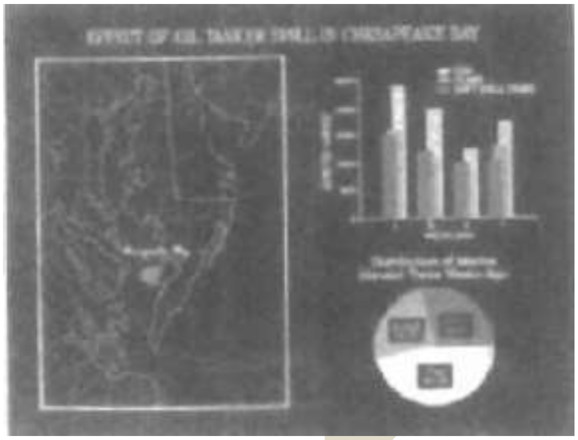
1.1.2 Presentation Graphics

Another major application area is presentation graphics, used to produce illustrations for reports or to generate 35-mm slides or transparencies for use with projectors. Presentation graphics is commonly used to summarize financial, statistical, Mathematical, scientific, and economic data for research reports, managerial reports, consumer information bulletins, and other types of reports. Workstation devices and service bureaus exist for converting screen displays into 35-mm slides or overhead transparencies for use in presentations. Typical examples of

Presentation graphics are bar charts, line graphs, surface graphs, pie charts, and other displays showing relationships between multiple parameters. In below Figure gives examples of two-dimensional graphics combined with geographical information. This illustration shows three color coded bar charts combined

Onto one graph and a pie chart with three sections. Similar graphs and charts can be displayed in three dimensions to provide additional information. Three-dimensional graphs are sometime used simply for effect; they can provide a more dramatic or more attractive presentation of data relationships.

(Fig: Two dimensional bar chart and pie chart)



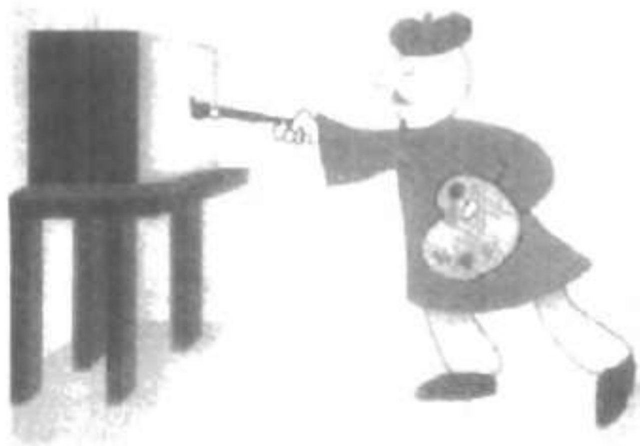
1.1.3 Computer Art

Computer graphics methods are widely used in both fine art and commercial art applications. Artists use a variety of computer methods, including special-purpose hardware, artist's paintbrush (such as Lumens), other paint packages (such as Pixel paint and Super paint), specially developed software, symbolic mathematics packages (such as Mathematics), CAD packages, desktop publishing software, and animation packages that provide facilities for designing object shapes and specifying object motions.

Fine artists use a variety of other computer technologies to produce images. For many applications of commercial art (and in motion pictures and other applications), photorealistic techniques are used to render images of a product. A common graphics method employed in many commercials is morphing,

where one object is transformed (metamorphosed) into another. This method has been used in TV commercials to turn an oil can into an automobile engine, an automobile into a tiger, a puddle of water into a tiger, and one person's face into another

face



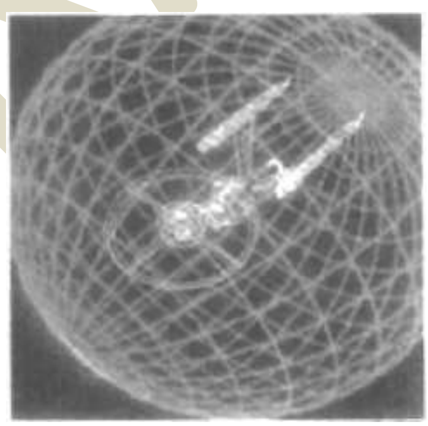
(Fig: Cartoon drawing produced with a paintbrush program)

In above figure illustrates the basic idea behind a paintbrush program that allows artists to "paint" pictures on the screen of a video monitor. Actually, the picture is usually painted electronically on a graphics tablet (digitizer) using a stylus, Which can simulate different brush strokes, brush widths, and colors. A paintbrush program was used to create the characters.

1.1.4 Entertainment

Computer graphics methods are now commonly used in making motion picture**s,** music videos, and television shows. Sometimes the graphics scenes are displayed by themselves, and sometimes graphics objects are combined with the actors and live scenes.

A graphics scene generated for the movie Star Trek-the Wrath of Khan is shown in below figure.



(Fig: Graphics developed for the paramount picture movie)

The planet and spaceship are drawn in wire fame form and will be shaded with rendering methods to produce solid surfaces. Many TV series regularly employ computer graphics methods.

Music videos use graphics in several ways. Graphics objects can be combined with the live action, or graphics and image processing techniques can be used to produce a transformation of one person or object into another (morphing). An example of morphing is shown in the sequence of scenes in below Figure, produced for the David Byme video She's Mad.

(Examples of morphing)

1.1.5 Education and Training

Computer-generated models of physical, financial, and economic systems are often used as educational aids. Models of physical systems, physiological systems, population trends, or equipment, such as the color coded diagram in below figure, can help trainees to understand the operation of the system. For some training applications, special systems are designed. Examples of Such specialized systems are the simulators for practice sessions or training of ship captains, aircraft pilots, heavy-equipment operators, and air traffic control personnel. Some simulators have no video screens**;** for example, a flight simulator with only a control panel for instrument flying. But most simulators provide graphics screens for visual operation.

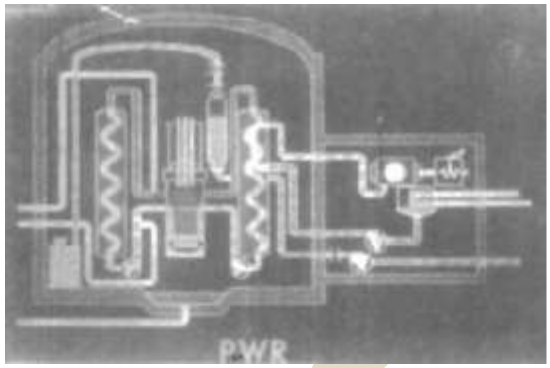


Fig: color coded diagram

1.1.6 Visualization

Scientists, engineers, medical personnel, business analysts, and others often need to analyze large amounts of information or to study the behavior of certain processes. Numerical simulations carried out on supercomputers frequently produce data files containing thousands and even millions of data values. initially,

Satellite cameras and other sources are amassing large data files faster than they can be interpreted. Scanning these large sets of number to determine trends and relationships is a tedious and ineffective process. But if the data are converted to a visual form, the trends and patterns are often immediately apparent. In below Figure shows an example of a large data set that has been converted to a color-coded display of relative heights above a ground plane.

Once we have plotted the density values in this way, we can see easily the overall pattern of the data. Producing Graphical representations for scientific, engineering, and medical data sets and processes are generally referred to as scientific visualization**.** And the term business visualization is used in connection with data sets related to commerce, industry, and other nonscientific areas. There are many different kinds of data sets, and effective visualization schemes depend on the characteristics of the data. A collection of data can contain scalar values, vectors, higher-order tensors, or any combination of these data types.

Data sets can be two-dimensional or three-dimensional. Color coding is just one way to visualize a data set. Additional techniques include contour plots, graphs and charts, surface renderings, and visualizations of volume interiors. In addition, image processing techniques are combined with computer graphics to produce many of the data visualizations.

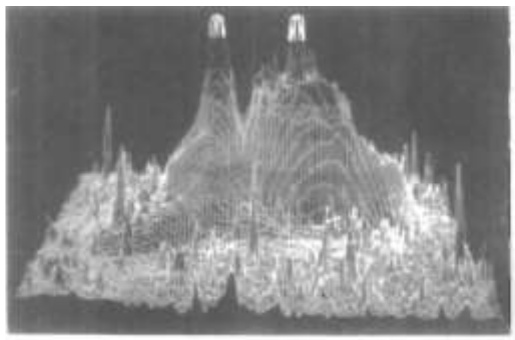


Fig: A color coded path with 16 million density points

1.1.7 Image Processing

Although methods used in computer graphics and Image processing overlap, the two areas with fundamentally different operations. In computer graphics, a computer is used to create a picture. Image processing, on the other hand. Applies techniques to modify or interpret existing picture, such as photographs and TV scans. Two principal applications of image processing are (1) improving picture quality and (2) machine perception of visual information, as used in robotics. To apply image processing methods, we first digitize a photograph or other Picture into an image file. Then digital methods can be applied to rearrange picture parts, to enhance color separations, or to improve the quality of shading. An example of the application of image processing methods to enhance the quality of a picture is shown in Fig. 1-70.

These techniques are used extensively in commercial art applications that involve the retouching and rearranging of sections of photographs and other artwork. Similar methods are used to analyze satellite photos of the earth and photos of galaxies. Medical applications also make extensive use of image processing techniques for picture enhancements, in tomography and in simulations of operations.

Tomography is a technique of X-ray photography that allows cross-sectional views of physiological systems to be displayed. Both computed X-rav tomography (CT) and position emission tomography (PET) use projection methods to reconstruct cross sections from digital data.

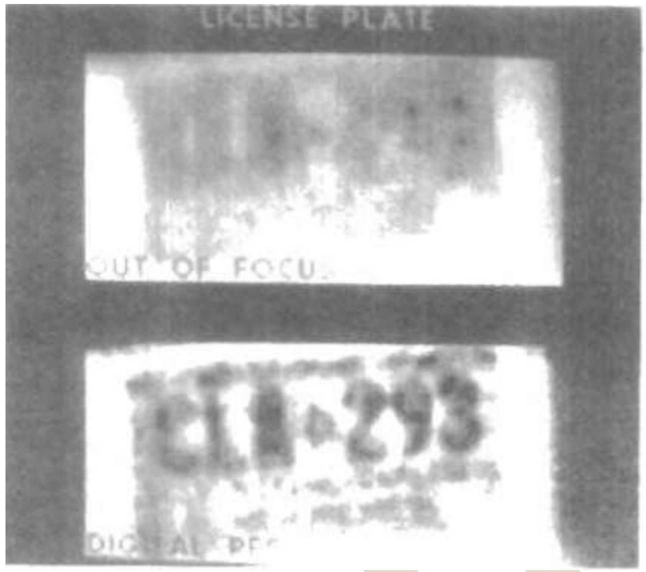


Fig. 1-70 (the blurred photograph of a license plate)

1.1.8 Graphical User Interfaces

It is common now for software packages to provide a graphical interface. A major component of a graphical interface is a window manager that allows a user to display multiple-window areas. Each window can contain a different process that can contain graphical or no graphical displays. To make a particular window active, we simply click in that window using an interactive pointing device. Interfaces also display menus and icons for fast selection of processing options or parameter values.

An icon is a graphical symbol that is designed to look

Like the processing option it represents. The advantages of icons are that they take up less screen space than corresponding textual descriptions and they can be understood more quickly if well designed. Menus contain lists of textual descriptions and icons.

Fig: A Graphical user interface showing multiple window areas, menus and icons.

In above figure illustrates a typical graphical interface, containing a window manager, menu displays, and icons. In this example, the menus allow selection of processing options, color values, and graphics parameters. The icons represent options for painting, drawing, zooming, typing text strings, and other operations connected with picture construction.

UNIT I

POSSIBLE QUESTIONS

2 MARKS

1. What is computer graphics?
2. What are the basic elements used in computer graphics?
3. Define texture.
4. What is Computer Aided Design?
5. Mention some applications of computer graphics.

POSSIBLE QUESTIONS

8 MARKS

1. Explain the basic elements of computer graphics.
2. Describe about applications of computer graphics.
3. Write a note on computer graphics.

UNIT-II

SYLLABUS

Graphics Hardware : Architecture of Raster and Random scan display devices, input/output devices.

Graphics Hardware

Graphics hardware is computer hardware that generates computer graphics and allows them to be shown on a display, usually using a graphics card (video card) in combination with a device driver to create the images on the screen.

Types

Graphics Cards

The most important piece of graphics hardware is the graphics card, which is the piece of equipment that renders out all images and sends them to a display. There are two types of graphics cards: integrated and dedicated. An integrated graphics card, usually by Intel to use in their computers, is bound to the motherboard and shares RAM(Random Access Memory) with the CPU, reducing the total amount of RAM available. This is undesirable for running programs and applications that use a large amount of video memory. A dedicated graphics card has its own RAM and Processor for generating its images, and does not slow down the computer. Dedicated graphics cards also have higher performance than integrated graphics cards. It is possible to have both[2] dedicated and integrated graphics, however once a dedicated graphics card is installed, the integrated card will no longer function until the dedicated card is removed.

Parts of a Graphics Card

The GPU, or graphics processing unit, is the unit that allows the graphics card to function. It performs a large amount of the work given to the card. The majority of video playback on a computer is controlled by the GPU. Once again, a GPU can be either integrated or dedicated.

Video Memory is built in RAM installed on the video card that provides the graphics card with its own RAM and allows it to run smoothly without having to take resources from the computer. However, if one uses an integrated graphics card, the chip will take resources from the computer as it does not have its own built in memory. That is how it is made.

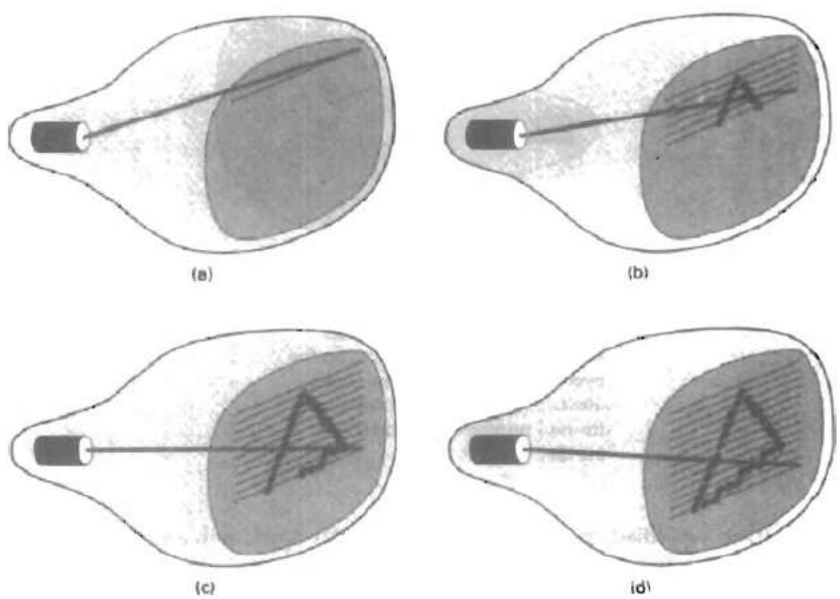
Display Drivers

A display driver is a piece of software which allows your graphics hardware to communicate with your operating system. Drivers in general allow your computer to utilize parts of itself, and without them, the machine would not function. This is because usually a graphics device communicates in its own language, which is more sophisticated, and a computer communicates in its own language, which largely deals with general commands. Therefore, a driver is required to translate between the two, and convert general commands into specific commands, and vice versa, so that each of the devices can understand the instructions and results.

ARCHITECTURE OF RASTER AND RANDOM SCAN DISPLAY DEVICES

Raster-Scan Displays

The most common type of graphics monitor employing a CRT is the raster-scan display, based on television technology. In a raster-scan system, the electron beam is swept across the screen, one row at a time from top to bottom. As the electron beam moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.



Picture definition is stored in a memory area called the refresh buffer or frame buffer. This memory area holds the set of intensity values for all the screen points. Stored intensity values are then retrieved from the refresh buffer and "painted" on the screen one row (scan line) at a time (in above Figure**).** Each screen point is referred to as a pixel or pel (shortened forms of picture element). The capability of a raster-scan system to store intensity information for each screen point makes it well suited for the realistic display of scenes containing subtle shading and color patterns. Home television sets and printers are examples of other systems using raster-scan methods.

Intensity range for pixel positions depends on the capability of the raster system. In a simple black-and-white system, each screen point is either on or off, so only one bit per pixel is needed to control the intensity of screen positions. For

a bi-level system, a bit value of 1 indicates that the electron beam is to be turn4 on at that position, and a value of 0 indicates that the beam intensity is to be off. Additional bits are needed when color and intensity variations can be displayed. Up to 24 bits per pixel are included in high-quality systems, which can require several megabytes of storage for the frame buffer, depending on the resolution of the system. A system with 24 bits per pixel and a screen resolution of 1024 by 1024 requires 3 megabytes of storage for the frame buffer. On a black-and-white system with one bit per pixel, the frame buffer is commonly called a bitmap. For systems with multiple bits per pixel, the frame buffer is often referred to as a pixmap.

Refreshing on raster-scan displays is carried out at the rate of 60 to 80 frames per second, although some systems are designed for higher refresh rates. Sometimes, refresh rates are described in units of cycles per second, or Hertz (Hz), where a cycle corresponds to one frame. Using these units, we would describe

a refresh rate of *60* frames per second as simply 60 Hz. At the end of each scan line, the electron beam returns to the left side of the screen to begin displaying the next scan line. The return to the left of the screen, after refreshing each scan line, is called the horizontal retrace of the electron beam. And at the end of

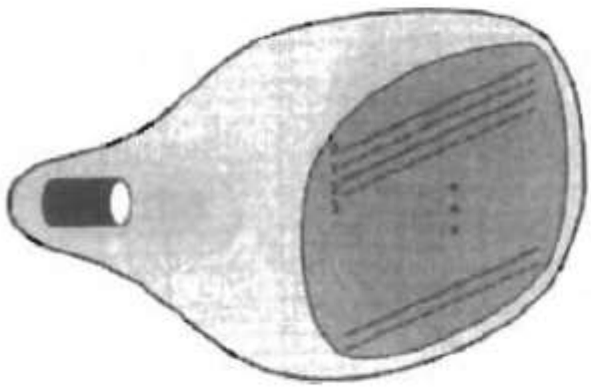
each frame (displayed in 1/80th to 1/60th of a second), the electron beam returns (vertical retrace) to the top left comer of the screen to begin the next frame.

Random-Scan Displays

When operated as a random-scan display unit, a CRT has the electron beam directed only to the parts of the screen where a picture is to be drawn. Random scan monitors draw a picture one line at a time and for this reason are also referred to as vector displays *(or* stroke-writing or calligraphic displays). The

Component lines of a picture can be drawn and refreshed by a random-scan system in any specified order which is shows in below figure.

(Fig: Interlacing scan lines on the raster scan system)



A pen plotter operates in a similar way and is an example of a random-scan, hard-copy device. Refresh rate on a random-scan system depends on the number of lines to be displayed. Picture definition is now stored as a set of line drawing commands in

an area of memory r e f e d to as the refresh display file. Sometimes the refresh display file is called the display list, display program, or simply the refresh buffer. To display a specified picture, the system cycles through the set of commands

in the display file, drawing each component line in turn. After all line drawing commands have been processed, the system cycles back to the first line command in the list. Random-scan displays are designed to draw all the component lines of a picture 30 to 60 times each second. High quality vector systems are capable of handling approximately 100,000 "short" lines at this refresh rate. When a small set of lines is to be displayed, each refresh cycle is delayed to avoid refresh rates greater than 60 frames per second. Otherwise, faster refreshing of the set of lines could bum out the phosphor.

Random-scan systems are designed for line drawing applications and cannot display realistic shaded scenes. Since picture definition is stored as a set of line drawing instructions and not as a set of intensity values for all screen points, vector displays generally have higher resolution than raster systems. Also, vector displays produce smooth line drawings because the CRT beam directly follows the line path. A raster system, in contrast, produces jagged lines that are plotted as discrete point sets.

INPUT DEVICES:

Input devices

Various devices *are* available for data input on graphics workstations. Most systems have a keyboard and one or more additional devices specially designed for iterative input. These include a mouse, trackball, space ball, joystick, digitizers, dials, and button boxes. Some other inputs devices are used In particular applications are data gloves, touch panels, image scanners, and voice systems.

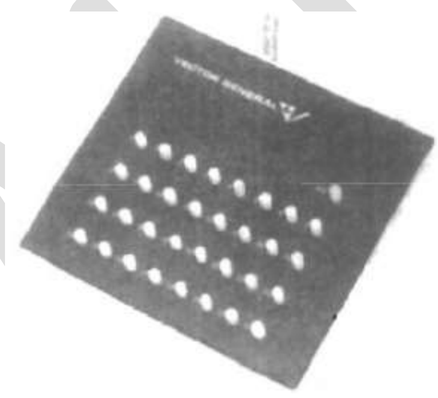
Keyboards

An alphanumeric keyboard on a graphics system is used primarily as a device for entering text strings. The keyboard is an efficient device for inputting such non graphic data as picture labels associated with a graphics display. Keyboards can also be provided with features to facilitate entry of screen coordinates, menu selections, or graphics functions. Cursor-control keys and function keys are common features on general purpose keyboards. Function keys allow users to enter frequently used operations in a single keystroke, and cursor-control keys can be used to select displayed objects or coordinate positions by positioning the screen cursor. Other types of cursor-positioning devices, such as a trackball or joystick, are included on some keyboards. Additionally, a numeric keypad is, often included on the keyboard for fast entry of numeric data. Typical examples of general-purpose keyboards are given in below Fig.

(Fig: Keyboard with removable palm rests)

For specialized applications, input to a graphics application may come from a set of buttons, dials, or switches that select data values or customized graphics operations. In below Figure gives an example of a button box and a set of input dials. Buttons and switches are often used to input predefined functions, and dials are common devices for entering scalar values. Real numbers within some defined range are selected for input with dial rotations. Potenhometers are used to measure dial rotations, which are then converted to deflection voltages for cursor movement.

(Fig: A Button box)



x

Mouse

A mouse is small hand-held box used to position the screen cursor. Wheels or rollers on the bottom of the mouse can be used to record the amount and direction of movement. Another method for detecting mouse motion is with an optical sensor. For these systems, the mouse is moved over a special mouse pad that has a grid of horizontal and vertical lines. The optical sensor detects movement across the lines in the grid. Since a mouse can be picked up and put down at another position without change in cursor movement, it is used for making relative change in the position of the screen cursor. One, two, or three button usually included on the top of the mouse for signaling the execution of some operation, such as recording cursor position or invoking a function. Mast general-purpose graphics systems now include a mouse and a keyboard as the major input devices.

Additional devices can be included in the basic mouse design to increase the number of allowable input parameters. The Z mouse in Fig. includes three buttons, a thumbwheel on the side, a trackball on the top, and a standard mouse ball underneath. This design provides six degrees of freedom to select Input Devices spatial positions, rotations, and other parameters. With the Z mouse, we can pick up an object, rotate it, and move it in any direction, or we can navigate our viewing position and orientation through a three-dimensional scene. Applications of the Z mouse include virtual reality, CAD, and animation.

(Fig: Z- Mouse)

Trackball and Space ball

As the name implies, a trackball is a ball that can be rotated with the fingers or palm of the hand, as in Fig, to produce screen-cursor movement. Potentiometers, attached to the ball, measure the amount and direction of rotation. Trackballs are often mounted on keyboards or other devices such as the Z mouse.

(Fig: A Three Button Track Ball)

While a trackball is a two-dimensional positioning device, a space ball provides six degrees of freedom. Unlike the trackball, a space ball does not actually move. Strain gauges measure the amount of pressure applied to the space ball to provide input for spatial positioning and orientation as the ball is pushed or pulled in various directions. Space balls are used for three-dimensional positioning and selection operations in virtual-reality systems, modeling, animation, CAD, and other applications.

Joysticks

A joystick consists of a small, vertical lever (called the stick) mounted on a base that is used to steer the screen cursor around. Most joysticks select screen positions with actual stick movement; others respond to pressure on the stick. Some joysticks are mounted on a keyboard; others function as stand-alone units. The distance that the stick is moved in any direction from its center position corresponds to screen-cursor movement in that direction. Potentiometers mounted at the base of the joystick measure the amount of movement, and springs return the stick to the center position when it is released. One or more buttons can be programmed to act as input switches to signal certain actions once a screen position has been selected.

In another type of movable joystick, the stick is used to activate switches that cause the screen cursor to move at a constant rate in the direction selected. Eight switches, arranged in a circle, are sometimes provided, so that the stick can select any one of eight directions for cursor movement. Pressure sensitive joysticks, also called isometric joysticks, have a non movable stick. Pressure on the stick is measured with strain gauges and converted to movement of the cursor in the direction specified.

Data Glove

From the below figure shows a data glove that can be used to grasp a "virtual" object. The glove is constructed with a series of sensors that detect hand and finger motions. Electromagnetic coupling between transmitting antennas and receiving antennas is used to provide information about the position and orientation of the hand. The transmitting and receiving antennas can each be structured as a set of three mutually perpendicular coils, forming a three-dimensional Cartesian coordinate system. Input from the glove can be used to position or manipulate objects in a virtual scene. A two-dimensional projection of the scene can be viewed on a video monitor, or a three-dimensional projection can be viewed with a headset.

(Fig: A virtual reality screen displayed on a 2-D video monitor with input from data glove and a space ball)

Digitizers

A common device for drawing, painting, or interactively selecting coordinate positions on an object is a digitizer. These devices can be used to input coordinate values in either a two-dimensional or a three-dimensional space. Typically, a digitizer is used to scan over a drawing or object and to input a set of discrete coordinate positions, which can be joined with straight-line segments to approximate the curve or surface shapes. One type of digitizer *is* the graphics tablet (also referred to as a data tablet), which is used to input two-dimensional coordinates by activating a hand cursor or stylus at selected positions on a flat surface. A hand cursor contains cross hairs for sighting positions, while a stylus is a pencil-shaped device that is pointed at positions on the tablet.

Many graphics tablets are constructed with a rectangular grid of wires embedded in the tablet surface. Electromagnetic pulses are generated in sequence along the wires, and an electric signal is induced in a wire coil in an activated stylus or hand cursor to record a tablet position. Depending on the technology, signal strength, coded pulses, or phase shifts can be used to determine the position on the tablet. - Acoustic (or sonic) tablets use sound waves to detect a stylus position. Either strip microphones or point microphones can be used to detect the sound emitted by an electrical spark from a stylus tip. The position of the stylus is calculated by timing the arrival of the generated sound at the different microphone positions. An advantage of two-dimensional acoustic tablets is that the microphones can be placed on any surface to form the "tablet" work area. This can be convenient for various applications, such as digitizing drawings in a book.

Three-dimensional digitizers use sonic or electromagnetic transmissions to word positions. One electromagnetic transmission method is similar to that used in the data glove: A coupling between the transmitter and receiver is used to compute the location of a stylus as it moves over the surface of an object.

Image Scanners

Drawings, graphs, color and black-and-white photos, or text can be stored for computer processing with an image scanner by passing an optical scanning mechanism over the information to be stored. The gradations of gray scale or color are then recorded and stored in an array. Once we have the internal representation of a picture, we can apply transformations to rotate, scale, or crop the picture to a particular screen area. We can also apply various image-processing methods to modify the array representation of the picture. For scanned text input, various editing operations can be performed on the stored documents. Some scanners are able to scan either graphical representations or text, and they Come in a variety of sizes and capabilities.

Touch Panels

As the name implies, touch panels allow displayed objects or screen positions to be selected with the touch of a finger. A typical application of touch panels is for the selection of processing options that are represented with graphical icons. Some systems, such as the plasma panels are designed with Touch screens. Other systems can be adapted for touch input by fitting a transparent device with a touch sensing mechanism over the video monitor screen. Touch input can be recorded using optical, electrical, or acoustical methods. Optical touch panels employ a line of infrared light-emitting diodes (LEDs)

Along one vertical edge and along one horizontal edge of the frame. The opposite vertical and horizontal edges contain light detectors. These detectors are used to record which beams are interrupted when the panel is touched.

The two crossing beams that are interrupted identify the horizontal and vertical coordinates of the screen position selected. Positions tin be selected with an accuracy of about ¼ inch With closely spaced LEDs, it is possible to break two horizontal or two vertical beams simultaneously. In this case, an average position between the two interrupted beams is recorded. The LEDs operate at infrared frequencies, so that the light is not visible to a user. The arrangement of LEDs in an optical touch panel that is designed to match the color and contours of the system to which it is to be fitted. An electrical touch panel is constructed with two transparent plates separated by a small distance. One of the plates is coated with a conducting material, and the other plate is coated with a resistive material. When the outer plate is touched, it is f o d into contact with the inner plate. This contact creates a voltage drop across the resistive plate that is converted to the coordinate values of the selected *screen* position.

In acoustical touch panels, high-frequency sound waves are generated in the horizontal and vertical directions across a glass plate. Touching the screen causes part of each wave to be reflected from the finger to the emitters. The screen position at the point of contact is calculated from a measurement of the time interval between the transmission of each wave and its reflection to the emitter.

Light Pens

In the below figure shows the design of one type of light pen. Such pencil-shaped devices are used to select screen positions by detecting the light coming from points on the CRT screen They are sensitive to *the* short burst of light emitted from the

Phosphor coating at the instant the electron beam strikes a particular point. Other Light sources, such as the background light in the room, are usually not detected by a light pen. An activated light pen, pointed at a spot on the screen as the electron beam lights up that spot, generates an electrical pulse that causes the coordinate position of the electron beam to be recorded. As with cursor-positioning devices, recorded Light-pen coordinates can be used to position an object or to select a processing option.

(Fig: Light pen)

Although Light pens are still with us, they are not as popular as they once were since they have several disadvantages command to other input devices that have been developed. For one, when a light pen is pointed at the screen, part of the screen image is obscured by the hand and pen. And prolonged use of the light pen can cause arm fatigue. Also, light pens require special implementations for some applications because they cannot detect positions within black areas. To be able to select positions in any screen area with a light pen, we must have some nonzero intensity assigned to each screen pixel. In addition, light pens. Sometimes give false readings due to background lighting in a room.

Voice Systems

Speech recognizers are used in some graphics workstations as input devices to accept voice commands The voice-system input can be used to initiate graphics operations or to enter data. These systems operate by matching an input against predefined dictionary of words and phrases. A dictionary is set up for a particular operator by having, the operator speak the command words to be used into the system. Each word is spoken Several times, and the system analyzes the word and establishes a frequency pattern for that word in the dictionary along with the corresponding function to be performed.

Later, when a voice command is given, the system searches the dictionary for a frequency-pattern match. Voice input is typically spoken into a microphone mounted on a headset, as in below Fig. The microphone is designed to minimize input of other background sounds. If a different operator is to use the

System, the dictionary must be reestablished with that operator's voice patterns. Voice systems have some advantage over other input devices, since the attention of the operator does not have to be switched from one device to another to enter a command.

(Fig : A speech recognition system)

Hard-Copy Devices

We can obtain hard-copy output for our images in several formats. For presentations or archiving, we can send image files to devices or service bureaus that will produce 35-mm slides or overhead transparencies. To put images on film, we can simply photograph a scene displayed on a video monitor. And we can put our pictures on paper by directing graphics output to a printer or plotter.

The quality of the pictures obtained from a device depends on dot size and the number of dots per inch, or Lines per inch, that can be displayed. To produce smooth characters in printed text strings, higher-quality printers shift dot positions so that adjacent dots overlap.

Printers produce output by either impact or nonimpact methods. Impact printers' press formed character faces against an inked ribbon onto the paper. A line printer is an example of an impact device, with the typefaces mounted on bands, chains, drums, or wheels. Nonimpact printers and plotters use laser techniques, ink-jet sprays, xerographic processes (as used in photocopying machines), electrostatic methods, and electro thermal methods to get images onto Paper.

Character impact printers often have a dot-matrix print head containing a rectangular array of protruding wire pins, with the number of pins depending on the quality of the printer. Individual characters or graphics patterns are obtained by retracting certain pins so that the remaining pins form the pattern to be printed. In a laser device, a laser beam mates a charge distribution on a rotating drum coated with a photoelectric material, such as selenium. Toner is applied to the drum and then transferred to paper.

Ink-jet methods produce output by squirting ink in horizontal rows across a roll of paper wrapped on a drum. The electrically charged ink stream is deflected by an electric field to produce dot-matrix patterns. A desktop ink-jet plotter with

a resolution of 360 dot per inch. An electrostatic device places a negative charge on the paper, one complete row at a time along the length of the paper. Then the paper is exposed to a toner. The toner is positively charged and so is attracted to the negatively charged areas, where it adheres to produce the specified output. Electro thermal methods use heat in a dot matrix print head to output patterns on heat sensitive paper.

OUTPUT DEVICES

- Monitors
- Graphic Plotter
- Printer

Monitors

Monitors, commonly called as **Visual Display Unit** (VDU), are the main output device of a computer. It forms images from tiny dots, called pixels that are arranged in a rectangular form. The sharpness of the image depends upon the number of pixels.

There are two kinds of viewing screen used for monitors.

- Cathode-Ray Tube (CRT)
- Flat-Panel Display

Cathode-Ray Tube (CRT) Monitor

The CRT display is made up of small picture elements called pixels. The smaller the pixels, the better the image clarity or resolution. It takes more than one illuminated pixel to form a whole character, such as the letter ‘e’ in the word help.

A finite number of characters can be displayed on a screen at once. The screen can be divided into a series of character boxes - fixed location on the screen where a standard character can be placed. Most screens are capable of displaying 80 characters of data horizontally and 25 lines vertically.

There are some disadvantages of CRT −

- Large in Size
- High power consumption

Flat-Panel Display Monitor

The flat-panel display refers to a class of video devices that have reduced volume, weight and power requirement in comparison to the CRT. You can hang them on walls or wear them on your wrists. Current uses of flat-panel displays include calculators, video games, monitors, laptop computer, and graphics display.

The flat-panel display is divided into two categories −

- **Emissive Displays** − Emissive displays are devices that convert electrical energy into light. For example, plasma panel and LED (Light-Emitting Diodes).
- **Non-Emissive Displays** − Non-emissive displays use optical effects to convert sunlight or light from some other source into graphics patterns. For example, LCD (Liquid-Crystal Device).

Printers

Printer is an output device, which is used to print information on paper.

There are two types of printers −

- Impact Printers
- Non-Impact Printers

Impact Printers

Impact printers print the characters by striking them on the ribbon, which is then pressed on the paper.

Characteristics of Impact Printers are the following −

- Very low consumable costs
- Very noisy
- Useful for bulk printing due to low cost
- There is physical contact with the paper to produce an image

These printers are of two types −

- Character printers
- Line printers

Character Printers

Character printers are the printers which print one character at a time.

These are further divided into two types:

- Dot Matrix Printer(DMP)
- Daisy Wheel

Dot Matrix Printer

In the market, one of the most popular printers is Dot Matrix Printer. These printers are popular because of their ease of printing and economical price. Each character printed is in the form of pattern of dots and head consists of a Matrix of Pins of size (5*7, 7*9, 9*7 or 9*9) which come out to form a character which is why it is called Dot Matrix Printer.

Advantages

- Inexpensive
- Widely Used
- Other language characters can be printed

Disadvantages

- Slow Speed
- Poor Quality

Daisy Wheel

Head is lying on a wheel and pins corresponding to characters are like petals of Daisy (flower) which is why it is called Daisy Wheel Printer. These printers are generally used for word-processing in offices that require a few letters to be sent here and there with very nice quality.

Advantages

- More reliable than DMP
- Better quality
- Fonts of character can be easily changed

Disadvantages

- Slower than DMP
- Noisy
- More expensive than DMP

Line Printers

Line printers are the printers which print one line at a time.

These are of two types −

- Drum Printer
- Chain Printer

Drum Printer

This printer is like a drum in shape hence it is called drum printer. The surface of the drum is divided into a number of tracks. Total tracks are equal to the size of the paper, i.e. for a paper width of 132 characters, drum will have 132 tracks. A character set is embossed on the track. Different character sets available in the market are 48 character set, 64 and 96 characters set. One rotation of drum prints one line. Drum printers are fast in speed and can print 300 to 2000 lines per minute.

Advantages

- Very high speed

Disadvantages

- Very expensive
- Characters fonts cannot be changed

Chain Printer

In this printer, a chain of character sets is used, hence it is called Chain Printer. A standard character set may have 48, 64, or 96 characters.

Advantages

- Character fonts can easily be changed.
- Different languages can be used with the same printer.

Disadvantages

- Noisy

Non-impact Printers

Non-impact printers print the characters without using the ribbon. These printers print a complete page at a time, thus they are also called as Page Printers.

These printers are of two types −

- Laser Printers
- Inkjet Printers

Characteristics of Non-impact Printers

- Faster than impact printers
- They are not noisy
- High quality
- Supports many fonts and different character size

Laser Printers

These are non-impact page printers. They use laser lights to produce the dots needed to form the characters to be printed on a page.

Advantages

- Very high speed

- Very high quality output
- Good graphics quality
- Supports many fonts and different character size

Disadvantages

- Expensive
- Cannot be used to produce multiple copies of a document in a single printing

Inkjet Printers

Inkjet printers are non-impact character printers based on a relatively new technology. They print characters by spraying small drops of ink onto paper. Inkjet printers produce high quality output with presentable features.

They make less noise because no hammering is done and these have many styles of printing modes available. Color printing is also possible. Some models of Inkjet printers can produce multiple copies of printing also.

Advantages

- High quality printing
- More reliable

Disadvantages

- Expensive as the cost per page is high
- Slow as compared to laser printer

POSSIBLE QUESTIONS

PART-B

2 MARKS

1. What is Graphics hardware?
2. Mention some Input devices and Output devices.
3. What is Raster Scan systems?

PART-C

6 MARKS

1) Discuss about Random Scan Systems.
2) Discuss about Raster Scan Systems.
3) Explain in detail about Digitizers and Joysticks.
4) Describe Impact Printer and its types.
5) Explain in detail about image scanners and mouse.
6) Explain hard copy device in details
7) Explain in detail about any 4 input devices with its operations.
8) Explain about Character printer and Inkjet printer
9) Explain in detail about image scanners and Voice Systems.
10) Explain in detail about any 4 input devices with its operations
11) Describe Printer and its types.

UNIT III

SYLLABUS

Fundamental Techniques in Graphics : Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

FUNDAMENTAL TECHNIQUES IN GRAPHICS

RASTER SCAN LINE

Line drawing algorithm

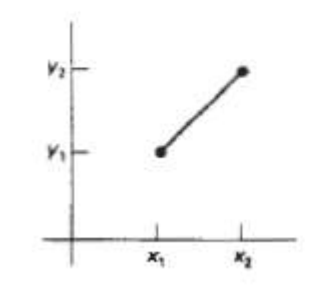
The Cartesian slope-intercept equation for a straight line is

$y = m \cdot x + b$ (3-1) with m representing the slope of the line and b as they intercept. Given that the two endpoints of a h e segment are specified at positions (*x*,, y,) and (*x*, y), as shown in Fig. 3-3, we can determine values for the slope m and y intercept b with the following calculations:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \qquad (3\text{-}2)$$

$$b = y_1 - m \cdot x_1 \qquad (3\text{-}3)$$

Fig: Line path between end point position (x_1, y_1) and (x_2, y_2).



Algorithms for displaying straight h e s are based on the line equation 3-1 and the calculations given in Eq. 3-2 and 3-3. For any given x interval Δx along a line, we can compute the corresponding y interval Δy from Eq. 3-2 as,

$$\Delta y = m \, \Delta x \qquad (3\text{-}4)$$

Similarly, we can obtain the x interval Δx corresponding to a specified as Δy

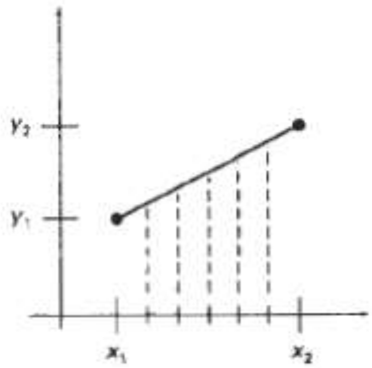
$$\Delta x = \frac{\Delta y}{m} \qquad (3\text{-}5)$$

These equations form the basis for determining deflection voltages in analog devices. For lines with slope magnitudes $|m| < 1$, Δx can be set proportional to a Δy as calculated from the equation 3-4 small horizontal deflection voltage and the corresponding vertical deflection is then set proportional to *Ay* as calculated from Eq. 3-4. For lines whose slopes have magnitudes 1 *m* I > 1, Δy can be set proportional to a small vertical deflection voltage with the corresponding horizontal deflection voltage set proportional to Δx, calculated from Eq. 3-5. For lines with *m* = 1, $\Delta x = \Delta y$ and the horizontal and vertical deflections voltages are equal. In each case, a smooth line with slope m is

generated between the specified endpoints. On raster systems, lines are plotted with pixels, and step sizes in the horizontal and vertical directions are constrained by pixel separations. That is, we must "sample" a line at discrete positions and determine the nearest pixel to the line at each sampled position. This scan conversion process for straight lines is illustrated in below Fig**,** for a near horizontal line with discrete sample positions

along the ***x*** axis.

Fig: Straight line segment with five sampling positions along the x axis between x_1 and x_2.



Circle generating algorithm

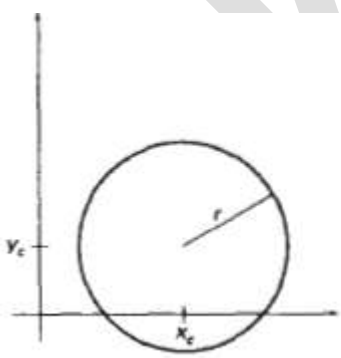
Since the circle is a frequently used component in pictures and graphs, a procedure for generating either full circles or circular arcs is included in most graphics packages. More generally, a single procedure can be provided to display either circular or elliptical curves.

Properties of Circles

A circle is defined as the set of points that are all at a given distance r from a center position (x, y) (Fig.1**).** This distance relationship is expressed by the Pythagorean theorem in Cartesian coordinates as

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \qquad (3\text{-}24)$$

Fig: 1 Circle with central coordinate (x_c, y_c) and radius r.



We could use this equation to calculate the position of points on a circle circumference by stepping along the x axis in unit steps from $x_c - r$ to $x_c + r$ and calculating the corresponding y values at each position as

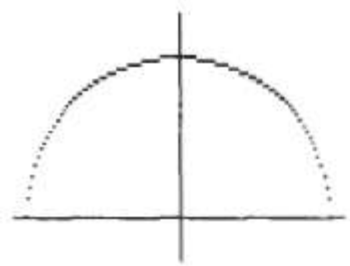
$$y = y_c \pm \sqrt{r^2 - (x_c - x)^2} \qquad (3\text{-}25)$$

But this is not the best method for generating a circle. One problem with this approach is that it involves considerable computation at each step. Moreover, the spacing between plotted pixel positions is not uniform, as demonstrated in below Figure2**.** We could adjust the spacing by interchanging ***x*** and y (stepping through y values and calculating x values) whenever the absolute value of the slope of the circle is greater than 1. But this simply increases the computation and processing required by the algorithm.

Another way to eliminate the unequal spacing shown in Fig. below Figure2 is to calculate points along the circular boundary using polar coordinates r and θ (Fig.**1).** Expressing the circle equation in parametric polar form yields the pair of equations

$$x = x_c + r\cos\theta$$
$$y = y_c + r\sin\theta \qquad (3\text{-}26)$$

Fig2:Positive half of a circle



When a display is generated with these equations using a fixed angular step size, a circle is plotted with equally spaced points along the circumference. The step size chosen for ***8*** depends on the application and the display device. Larger angular separations along the circumference can be connected with straight line segments to approximate the circular path. For a more continuous boundary on a raster display, we can set the step size at l/r. This plots pixel positions that are approximately one unit apart.

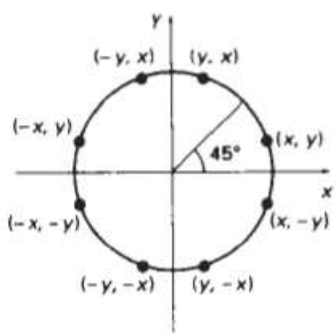
Computation can be reduced by considering the symmetry of circles. The shape of the circle is similar in each quadrant. We can generate the circle section in (he second quadrant of the xy plane by noting that the two circle sections are symmetric with respect to they axis. And circle sections in the third and fourth

quadrants can be obtained from sections in the first and second quadrants by considering symmetry about the *x* axis. We can take this one step further and note that there is also symmetry between octants. Circle sections in adjacent *oc*tants within one quadrant are symmetric with respect to the $45°$ line dividing the two octants. These symmetric conditions are illustrated in Fig.3, where a point at position *(x,* y) on a one-eighth circle sector is mapped into the seven circle points in the other octants of the xy plane. Taking advantage of the circle symmetry in this way we can generate all pixel positions around a circle by calculating

only the points within the sector from $x = 0$ to x = y.

Determining pixel positions along a circle circumference using either Eq. 3-24 or Eq. 3-26 still requires a good deal of computation time. The Cartesian equation 3-24 involves multiplications and square-root calculation, while the parametric equations contain multiplications and trigonometric calculations. More efficient circle algorithms are based on incremental calculation of decision parameters, as in the Bresenham line algorithm, which involves only simple integer operations,

Fig3: Symmetry of a circle. Calculation pf a circle point (x,y) in one octant yields circle points shown for the other seven octant.



Ellipse-Generating Algorithms

Ellipse-Generating Algorithms An ellipse is an elongated circle. Therefore, elliptical curves can be generated by modifying circle-drawing procedures to take into account the different dimensions of an ellipse along the major and minor axes.

Properties of ellipses An ellipse can be given in terms of the distances from any point on the ellipse to two fixed positions called the foci of the ellipse. The sum of these two distances is the same values for all points on the ellipse. If the distances to the two focus positions from any point p=(x,y) on the ellipse are labeled d1 and d2, then the general equation of an ellipse can be stated as d1+d2=constant Expressing distances d1 and d2 in terms of the focal coordinates F1=(x1,y2) and F2=(x2,y2) sqrt((x-x1) 2 +(y-y1) 2)+sqrt((x-x2) 2 +(y-y2) 2)=constant By squaring this equation isolating the remaining radical and squaring again. The general ellipse equation in the form

Ax2 +By2 +Cxy+Dx+Ey+F=0 The coefficients A,B,C,D,E, and F are evaluated in terms of the focal coordinates and the dimensions of the major and minor axes of the ellipse. The major axis is the straight line segment extending from one side of the ellipse to the other through the foci. The minor axis spans the shorter dimension of the ellipse, perpendicularly bisecting the major axis at the halfway position (ellipse center) between the two foci. An interactive method for specifying an ellipse in an arbitrary orientation is to input the two foci and a point on the ellipse boundary. Ellipse equations are simplified if the major and minor axes are oriented to align with the coordinate axes. The major and minor axes oriented parallel to the x and y axes parameter rx for this example labels the semi major axis and parameter ry labels the semi minor axis

((x-xc)/rx) 2 +((y-yc)/ry) 2 =1 Using polar coordinates r and θ, to describe the ellipse in Standard position with the parametric equations x=xc+rxcos θ

y=yc+rxsin θ Angle θ called the eccentric angle of the ellipse is measured around the perimeter of a bounding circle. We must calculate pixel positions along the elliptical arc throughout one quadrant, and then we obtain positions in the remaining three quadrants by symmetry

Midpoint ellipse Algorithm The midpoint ellipse method is applied throughout the first quadrant in two parts. The below figure show the division of the first quadrant according to the slope of an ellipse with rx<ry.

<u>THICK PRIMITIVES</u>

Drawing with Thick Primitives • How do we thicken the line stroke width? • Ideas: – Place the center of the circular “brush” on the pixel – Place the upper corner of the square “marker” on the pixel (issues of orientation) – Then do scan conversion algorithm

Three Basic Methods

1. Column Replication – Use >1 pixel per col/row

2. Trace brush outline across 1-pixel primitive

3. Trace two copies, t apart, and fill in

Column(Row) Replication

Idea: duplicate pixels in – Columns, when -1 < slope < 1 – Rows, otherwise

• Thickness t is from primitive's boundaries perpendicular to its tangent

• What happens for even t?

• Issues when lines meet at angles, when octants merge, brightness for sloped lines, etc.

Moving the Pen

Example: – a rectangular pen – Center or corner follows scan algorithm

• How to implement? –

Idea 1: fill the box at each point –

Problem: pixels get colored more than once –

Idea 2: fill by using a span of the pen primitive at each step

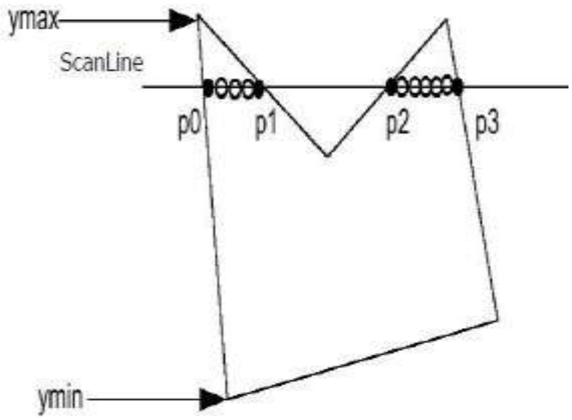
POLYGON FILLING:

Polygon is an ordered list of vertices as shown in the following figure. For filling polygons with particular colors, you need to determine the pixels falling on the border of the polygon and those which fall inside the polygon. In this chapter, we will see how we can fill polygons using different techniques.

Scan Line Algorithm

This algorithm works by intersecting scanline with polygon edges and fills the polygon between pairs of intersections. The following steps depict how this algorithm works.

Step 1 − Find out the Ymin and Ymax from the given polygon.



Step 2 − ScanLine intersects with each edge of the polygon from Ymin to Ymax. Name each intersection point of the polygon. As per the figure shown above, they are named as p0, p1, p2, p3.

Step 3 − Sort the intersection point in the increasing order of X coordinate i.e. (p0, p1), (p1, p2), and (p2, p3).

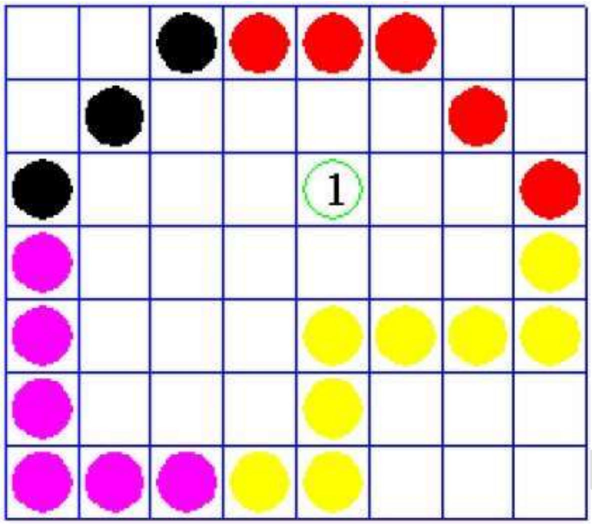
Step 4 − Fill all those pair of coordinates that are inside polygons and ignore the alternate pairs.

Flood Fill Algorithm

Sometimes we come across an object where we want to fill the area and its boundary with different colors. We can paint such objects with a specified interior color instead of searching for particular boundary color as in boundary filling algorithm.

Instead of relying on the boundary of the object, it relies on the fill color. In other words, it replaces the interior color of the object with the fill color. When no more pixels of the original interior color exist, the algorithm is completed.

Once again, this algorithm relies on the Four-connect or Eight-connect method of filling in the pixels. But instead of looking for the boundary color, it is looking for all adjacent pixels that are a part of the interior.



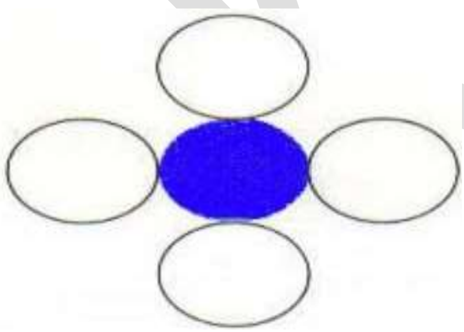
Boundary Fill Algorithm

The boundary fill algorithm works as its name. This algorithm picks a point inside an object and starts to fill until it hits the boundary of the object. The color of the boundary and the color that we fill should be different for this algorithm to work.

In this algorithm, we assume that color of the boundary is same for the entire object. The boundary fill algorithm can be implemented by 4-connected pixels or 8-connected pixels.

4-Connected Polygon

In this technique 4-connected pixels are used as shown in the figure. We are putting the pixels above, below, to the right, and to the left side of the current pixels and this process will continue until we find a boundary with different color.



Algorithm

Step 1 − Initialize the value of seed point (seedx, seedy), fcolor and dcol.

Step 2 − Define the boundary values of the polygon.

Step 3 − Check if the current seed point is of default color, then repeat the steps 4 and 5 till the boundary pixels reached.

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

Step 4 − Change the default color with the fill color at the seed point.

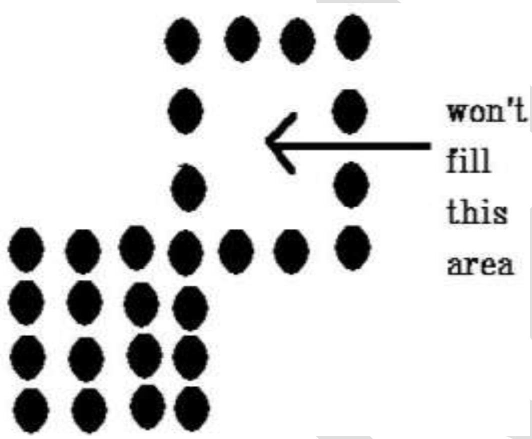
```
setPixel(seedx, seedy, fcol)
```

Step 5 − Recursively follow the procedure with four neighborhood points.

```
FloodFill (seedx – 1, seedy, fcol, dcol)
FloodFill (seedx + 1, seedy, fcol, dcol)
FloodFill (seedx, seedy - 1, fcol, dcol)
FloodFill (seedx – 1, seedy + 1, fcol, dcol)
```

Step 6 − Exit

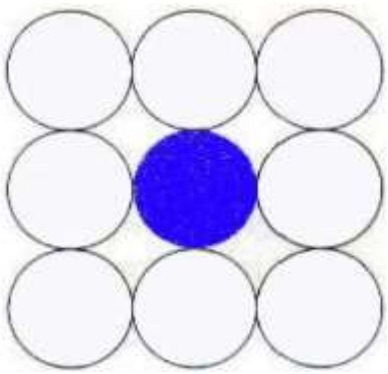
There is a problem with this technique. Consider the case as shown below where we tried to fill the entire region. Here, the image is filled only partially. In such cases, 4-connected pixels technique cannot be used.



8-Connected Polygon

In this technique 8-connected pixels are used as shown in the figure. We are putting pixels above, below, right and left side of the current pixels as we were doing in 4-connected technique.

In addition to this, we are also putting pixels in diagonals so that entire area of the current pixel is covered. This process will continue until we find a boundary with different color.



Algorithm

Step 1 − Initialize the value of seed point (seedx, seedy), fcolor and dcol.

Step 2 − Define the boundary values of the polygon.

Step 3 − Check if the current seed point is of default color then repeat the steps 4 and 5 till the boundary pixels reached

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

Step 4 − Change the default color with the fill color at the seed point.

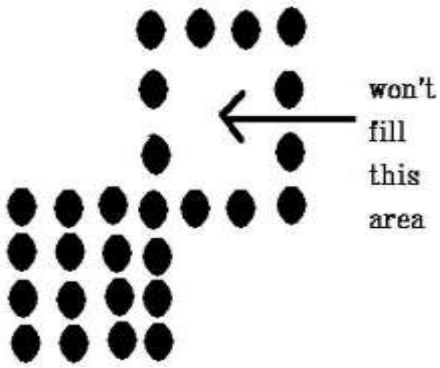
```
setPixel(seedx, seedy, fcol)
```

Step 5 − Recursively follow the procedure with four neighbourhood points

```
FloodFill (seedx – 1, seedy, fcol, dcol)
FloodFill (seedx + 1, seedy, fcol, dcol)
FloodFill (seedx, seedy - 1, fcol, dcol)
FloodFill (seedx, seedy + 1, fcol, dcol)
FloodFill (seedx – 1, seedy + 1, fcol, dcol)
FloodFill (seedx + 1, seedy + 1, fcol, dcol)
FloodFill (seedx + 1, seedy - 1, fcol, dcol)
FloodFill (seedx – 1, seedy - 1, fcol, dcol)
```

Step 6 − Exit

The 4-connected pixel technique failed to fill the area as marked in the following figure which won't happen with the 8-connected technique.



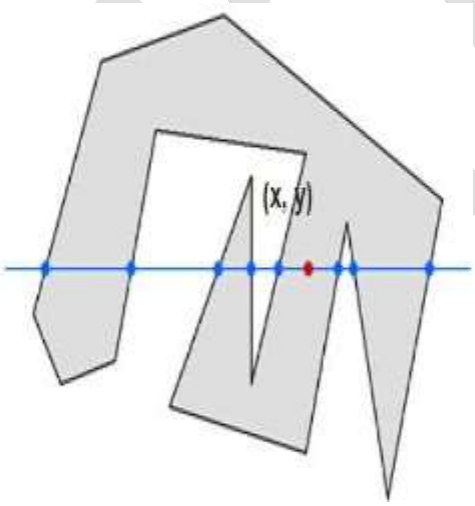
Inside-outside Test

This method is also known as **counting number method**. While filling an object, we often need to identify whether particular point is inside the object or outside it. There are two methods by which we can identify whether particular point is inside an object or outside.

- Odd-Even Rule
- Nonzero winding number rule

Odd-Even Rule

In this technique, we will count the edge crossing along the line from any point (x,y) to infinity. If the number of interactions is odd, then the point (x,y) is an interior point; and if the number of interactions is even, then the point (x,y) is an exterior point. The following example depicts this concept.

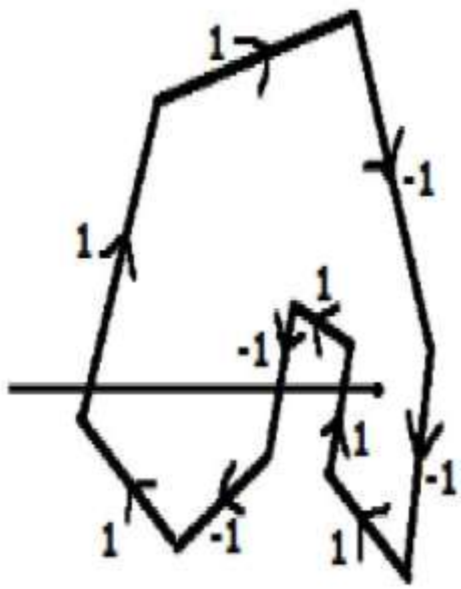


From the above figure, we can see that from the point (x,y), the number of interactions point on the left side is 5 and on the right side is 3. From both ends, the number of interaction points is odd, so the point is considered within the object.

Nonzero Winding Number Rule

This method is also used with the simple polygons to test the given point is interior or not. It can be simply understood with the help of a pin and a rubber band. Fix up the pin on one of the edge of the polygon and tie-up the rubber band in it and then stretch the rubber band along the edges of the polygon.

When all the edges of the polygon are covered by the rubber band, check out the pin which has been fixed up at the point to be test. If we find at least one wind at the point consider it within the polygon, else we can say that the point is not inside the polygon.



In another alternative method, give directions to all the edges of the polygon. Draw a scan line from the point to be test towards the left most of X direction.

- Give the value 1 to all the edges which are going to upward direction and all other -1 as direction values.
- Check the edge direction values from which the scan line is passing and sum up them.
- If the total sum of this direction value is non-zero, then this point to be tested is an **interior point,** otherwise it is an **exterior point**.

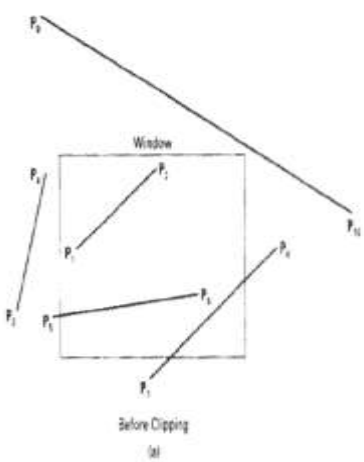
- In the above figure, we sum up the direction values from which the scan line is passing then the total is 1 – 1 + 1 = 1; which is non-zero. So the point is said to be an interior point.

LINE AND POLYGON CLIPPING ALGORITHM

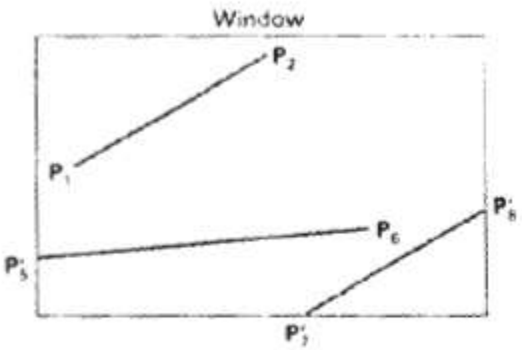
Line clipping

Figure6 illustrates possible relationships between line positions and a standard rectangular clipping region. A line clipping procedure involves several parts. First, we can test a given line segment to determine whether it lies completely inside the clipping window. If it does not, we try to determine whether it lies completely outside the window. Finally, if we cannot identify a line as completely inside or completely outside, we must perform intersection calculations with one or more clipping boundaries. We process lines through the "inside-outside" tests by checking the line endpoints. A line with both endpoints inside all clipping boundaries, such as the line from P_1 to P_2, is saved. A line with both endpoints outside any one of the clip boundaries (line $\overline{P_3P_4}$ in Fig.6) is outside the window.

Fig: 6 Line clipping against a rectangular shape window.



All other lines cross one or more clipping boundaries, and may require calculation of multiple intersection points. to minimize calculations, we try to devise clipping algorithms that can efficiently identify outside lines and reduce intersection calculations.



After Clipping
(b)

For a line segment with endpoints (x_1, y_1) and (x_2, y_2) and one or both endpoints outside the clipping rectangle, the parametric representation.

$$x = x_1 + u(x_2 - x_1)$$
$$y = y_1 + u(y_2 - y_1), \quad 0 \le u \le 1 \qquad (6\text{-}6)$$

could be used to determine values of parameter u for intersections with the clipping boundary coordinates. If the value of *u* for an intersection with a rectangle boundary edge is outside the range *0* to 1, the line does not enter the interior of the window at that boundary. If the value of *u* is with in the range from 0 to 1, the line segment does indeed cross into the clipping area. This method can be applied to each clipping boundary edge in turn to determine whether any part of the line segment is to be displayed. Line segments that are parallel to window edges can be handled as specia1 cases.

Clipping line segments with these parametric tests requires a good deal of

Computation and faster approaches to clipping are possible. A number of efficient line clippers have been developed, and we survey the major algorithms in the next section. Some algorithm are designed explicitly for two-dimensional

Pictures and some are easily adapted to three-dimensional applications.

Cohen-Sutherland line clipping

This is one of the oldest and most popular line-clipping procedures. Generally, the method speeds up the processing of line segments performing initial tests that reduce the number of intersections that must he calculated. Every line end Point in a picture is assigned a four-digit binary code, called a region code that identifies the location of the point relative to the boundaries of the clipping rectangle. Regions are set up in reference to the boundaries as shown in Fig.7**.** Each bit position in the region code is used to indicate one of the four relative coordinate positions of the point with respect to the clip window: to the left, right, top, or bottom. By numbering the bit positions in the region code as 1 through 4 from right to left, the coordinate regions can be correlated with the bit positions as

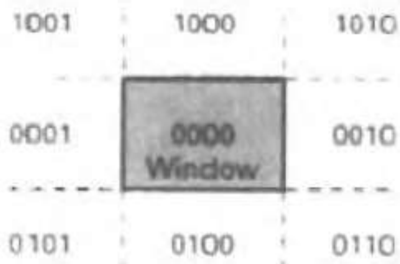
bit 1: left

bit 2: right

bit 3: below

bit 4: above

Fig7: Binary region codes assigned to line endpoints according to relative position with respect to the clipping rectangle.

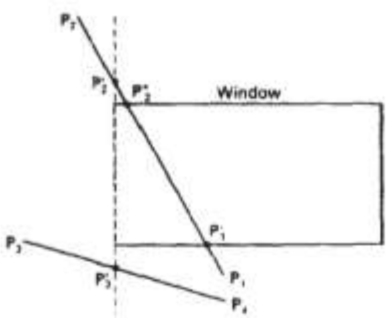


A value of 1 in any bit position indicates that the point is in that relative position; Otherwise, the bit position is set to 0. If a point is within the clipping rectangle, the region code is 0000. A point that is below and to the left of the rectangle has a region code of 0101. Bit values in the region code are determined by comparing endpoint Coordinate values (x, y) to the clip boundaries. Bit 1 is set to 1 if x < xw_{min},. The other three bit values can be determined using similar comparisons. For languages in which bit manipulation is possible, region-code bit values can be determined with the following two steps: (1) Calculate differences between endpoint coordinates and clipping boundaries. (2) Use the resultant sign bit of each difference calculation to set the corresponding value in the region code. Bit **1** is the sign bit of ***x*** - bit 2 is the sign bit of $xw_{max} - x$; bit 3 is the sign bit of $y - yw_{min}$; and bit 4 is the sign bit of $yw_{max} - y$.

Once we have established region codes for all line endpoints, we can quickly determine which lines are completely inside the clip window and which are clearly outside. Any lines that are completely contained within the window boundaries have a region code of 0000 for both endpoints, and we trivially accept these lines. Any lines that have a 1 in the same bit position in the region codes for each endpoint are completely outside the clipping rectangle, and we trivially reject these lines. We would discard the line that has a region code of 1001 for one endpoint and a code of 0101 for the other endpoint. Both endpoints of this line are left of the clipping rectangle, as indicated by the 1 in the first bit position of each region code. A method that can be used to test lines for total clipping is to perform the logical and operation with both region codes. If the result is not 0000, the line is completely outside the clipping region.

Lines that cannot be identified as completely inside or completely outside a Clip window by these tests are checked for intersection with the window boundaries. As shown in Fig.8, such lines may or may not cross into the window interior. We begin the clipping process for a line by comparing an outside endpoint to a clipping boundary to determine how much of the line can be discarded. Then the remaining part of the Line is checked against the other boundaries, and we continue until either the line is totally discarded or a section is found inside the window. We set up our algorithm to check line endpoints against clipping boundaries in the order left, right, bottom, top.

Fig8: Lines extending from one coordinate region to another may pass through the clip window, or they may intersect clipping boundaries without entering the window.



To illustrate the specific steps in clipping lines against rectangular boundaries using the Cohen-Sutherland algorithm, we show how the lines in Fig.8 could be processed. Starting with the bottom endpoint of the line from P_1 to P_2,

We check P, against the left, right, and bottom boundaries in turn and find that this point is below the clipping rectangle. We then find the intersection point Pi with the bottom boundary and discard the line section from P_1 to P_1'. The line now has been reduced to the section from P_1' to P_2. Since P_2 is outside the clip window, we check this endpoint against the boundaries and find that it is to the left of the window. Intersection point P_2' is calculated, but this point is above the window. So the final intersection calculation yields P_2'', and the line from P_1' to P_2'' is saved. This completes processing for this line, so we save this part and go on to the next line. Point P_3 in the next line is to the left of the clipping rectangle, so we determine the intersection P_3' and eliminate the line section from P_3 to P_3'. By checking region codes for the line section from P_3' to P_4, we find that the remainder of the line is below the clip window and can be discarded also.

Intersection points with a clipping boundary can be calculated using the

Slope-intercept form of the line equation. For a line with endpoint coordinates (x,y,) and (x2, y2), they coordinate of the intersection point with a vertical boundary can be obtained with the calculation.

$$y = y_1 + m(x - x_1) \qquad (6\text{-}7)$$

where the ***x*** value is set either to xw_{min} or to xw_{max}, and the slope of the line is calculated as $m = (y_2 - y_1)/(x_2 - x_1)$. Similarly, if we are looking for the intersection with a horizontal boundary, the ***x*** coordinate can be calculated as

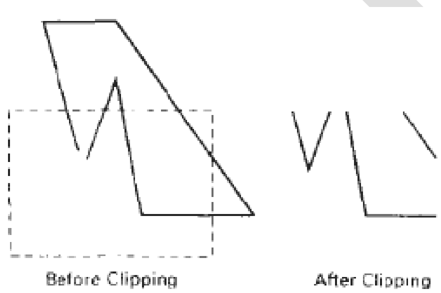
$$x = x_1 + \frac{y - y_1}{m} \qquad (6\text{-}8)$$

with y set either to yw_{min} or to yw_{max}.

Polygon clipping

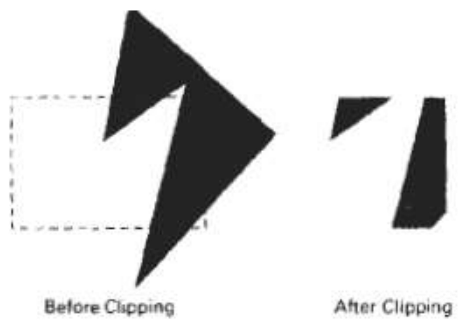
To clip polygons, we need to modify the line-clipping procedures discussed in the previous section. A polygon boundary processed with a line clipper may be displayed as a series of unconnected line segments (Fig.9), depending on the Orientation of the polygon to the clipping window.

Fig9: Display of a polygon processed by a line-dipping algorithm.



What we really want to display is a bounded area after clipping, as in Fig. 10. For polygon clipping, we require an algorithm that will generate one or more closed areas that are then scan converted for the appropriate area fill. The output of a polygon clipper should be a sequence of vertices that defines the clipped polygon boundaries.

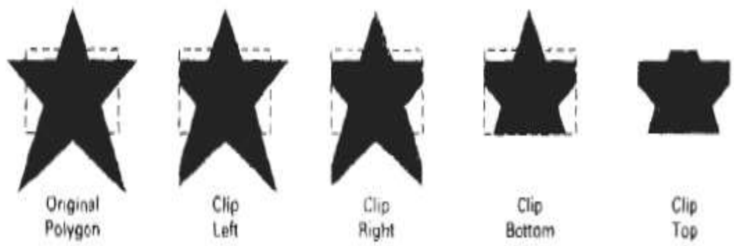
Fig10: display of a correctly clipped polygon.



Sutherland-Hodgeman polygon clipping

We can correctly clip a polygon by processing the polygon boundary as a whole against each window edge. This could be accomplished by processing all polygon vertices against each clip rectangle boundary in turn. Beginning with the initial set of polygon vertices, we could first clip the polygon against the left rectangle boundary to produce a new sequence of vertices. The new set of vertices could then k successively passed to a right boundary clipper, a bottom boundary clipper, and a top boundary clipper, as in Fig.11. At each step, a new sequence of output vertices is generated and passed to the next window boundary clipper.

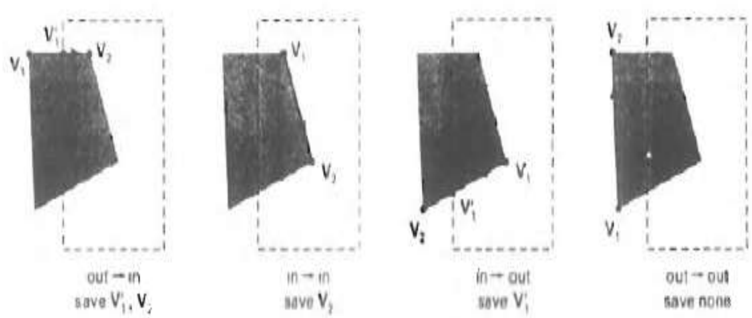
Fig11: Clipping a polygon against successive window boundaries.



There are four possible cases when processing vertices in sequence around the perimeter of a polygon. As each pair of adjacent polygon vertices is passed to a window boundary clipper, we make the following tests: (1) If the first vertex is

Outside the window boundary and the second vertex is inside, both the intersection point of the polygon edge with the window boundary and the second vertex are added to the output vertex list. *(2)* If both input vertices are inside the window boundary, only the second vertex is added to the output vertex list. (3) if the first vertex is inside the window boundary and the second vertex is outside, only the edge intersection with the window boundary is added to the output vertex list. (4) If both input vertices are outside the window boundary, nothing is added to the output list. These four cases are illustrated in Fig.12, for successive pairs of polygon vertices. Once all vertices have been processed for one clip window boundary; the output 1ist of vertices is clipped against the next window boundary.

Fig12: Successive processing of pairs of polygon vertices against the left window boundary.



2D GEOMETRIC TRANSFORMATION

Two dimensional geometric transformation

With the procedures for displaying output primitives and their attributes, we can create variety of pictures and graphs. In many applications, there is also a need for altering or manipulating displays. Design applications and facility layouts are created by arranging the orientations and sizes of the component parts of the scene. And animations are produced by moving the "camera" or the objects in a scene along animation paths. Changes in orientation, size, and shape are accomplished with geometric transformations that alter the coordinate descriptions of objects. The basic geometric transformations are translation, rotation, and scaling. Other transformations that are often applied to objects include reflection and shear. We first discuss methods for performing geometric transformations and then consider how transformation functions can be incorporated into graphics packages.

Basic transformation

Here, we first discuss general procedures for applying translation, rotation, and scaling parameters to reposition and resize two-dimensional objects. Then, we consider how transformation equations can be expressed in a more convenient matrix formulation that allows efficient combination of object transformations.

Translation

A translation is applied to an object by repositioning it along a straight-line path from one coordinate location to another. We translate a two-dimensional point by adding translation distances, t_x and t_y, to the original coordinate position **(x,** y) to move the point to a new position ***(x ' ,*** y'). (Fig.1).

$$x' = x + t_x, \qquad y' = y + t_y \qquad (5\text{-}1)$$

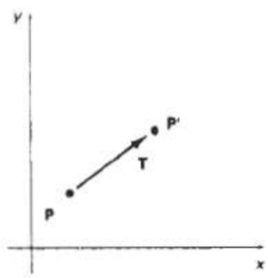
The translating and distance pair (t_x, t_y) is called a translation vector or shift vector. We can express the translation equations 5-1 as a single matrix equation by using column vectors to represent coordinate positions and the translation vector:

$$\mathbf{P} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{P}' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad (5\text{-}2)$$

This allows us to write the two-dimensional translation equations in the matrix form:

$$\mathbf{P}' = \mathbf{P} + \mathbf{T} \qquad (5\text{-}3)$$

Fig1: Translating a point from position P to the position p' with translation vector V.

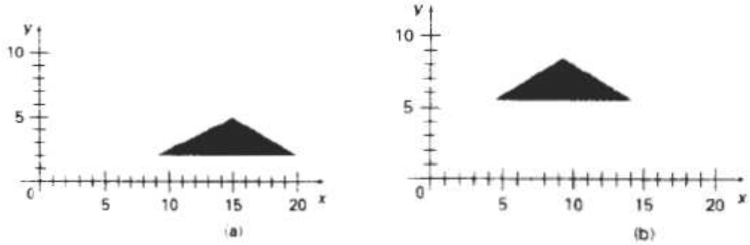


Sometimes matrix-transformation equations are expressed in terms of coordinate row vectors instead of column vectors. In this case, we would write the matrix representations as,

$\mathbf{P} = [x \; y]$ and $\mathbf{T} = [t_x \; t_y]$.

Translation is a rigid-body transformation that moves objects without deformation. That is, every point on the object is translated by the same amount. A straight Line segment is translated by applying the transformation equation 5-3 to each of the line endpoints and redrawing the line between the new endpoint positions. Polygons are translated by adding the translation vector to the coordinate position of each vertex and regenerating the polygon using the new set of vertex coordinates and the current attribute settings. Figure 2 illustrates the application of a specified translation vector to move an object from one position to another.

Fig2: Moving polygon from position (a) to position (b) with the translation vector.

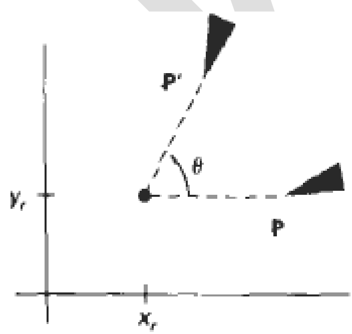


Similar methods are used to translate curved objects. To change the position of a circle or ellipse, we translate the center coordinates and redraw the figure in the new location. We translate other curves (for example, splines) by displacing the coordinate positions defining the objects, then we reconstruct the curve paths using the translated coordinate points.

Rotation

A two-dimensional rotation is applied to an object by repositioning it along a circular path in the xy plane. To generate a rotation, we specify a rotation angle θ and the position (x_r, y_r) of the rotation point (or pivot point) about which the object is to be rotated (Fig.3**).** Positive values for the rotation angle define counterclockwise Rotations about the pivot point, as in Fig.3, and negative values rotate objects in the clockwise direction. This transformation can also be described as a rotation about a rotation axis that is perpendicular to the xy plane and passes through the pivot point.

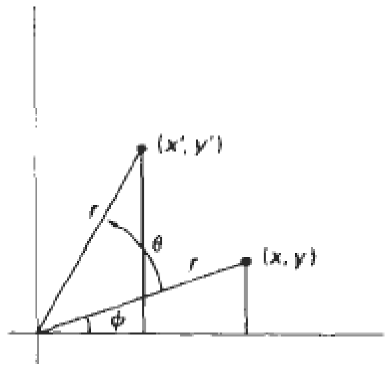
Fig 3: Rotation of an object through angle θ about the pivot point



We first determine the transformation equations for rotation of a point position P when the pivot point is at the coordinate origin. The angular and coordinate relationships of the original and transformed point positions are shown in Fig.4. In this figure r is the constant distance of the point from the origin, angle ϕ is the original angular position of the point from the horizontal, and θ is the rotation angle. Using standard trigonometric identities, we can express the transformed coordinates in terms of angles θ and ϕ as,

Fig 4: Rotation of a point from position (x, y) to position (x', y')

Through an angle θ relative to the coordinate origin.



$$x' = r\cos(\phi + \theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta$$
$$y' = r\sin(\phi + \theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \qquad (5\text{-}4)$$

The original coordinates of the point in polar coordinates are

$$x = r\cos\phi, \qquad y = r\sin\phi \qquad (5\text{-}5)$$

Substituting expressions 5-5 into 5-4, we obtain the transformation equations for rotating a point at position *(x*, y) through an angle 9 about the origin:

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta \qquad (5\text{-}6)$$

With the column-vector representations 5-2 for coordinate positions, we can write the rotation equations in the matrix form:

$$\mathbf{P}' = \mathbf{R} \cdot \mathbf{P} \qquad (5\text{-}7)$$

where the rotation matrix is

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad (5\text{-}8)$$

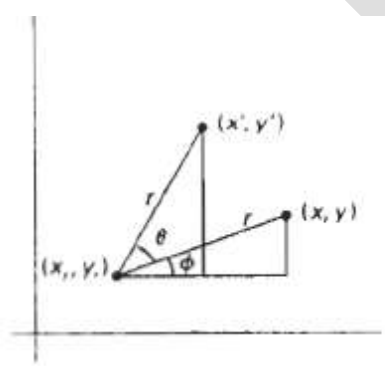
When coordinate positions are represented as row vectors instead of column vectors, the matrix product in rotation equation 5-7 is transposed so that the transformed row coordinate vector (x' y') is calculated as

$$\mathbf{P}'^T = (\mathbf{R} \cdot \mathbf{P})^T$$
$$= \mathbf{P}^T \cdot \mathbf{R}^T$$

Where $\mathbf{P}^T = [x\ y]$, and the transpose $\mathbf{R}^T$ of matrix R is obtained by interchanging rows and columns. For a rotation matrix, the transpose is obtained by simply changing the sign of the sine terms.

Rotation of a point about an arbitrary pivot position is illustrated in Fig. 5. Using the trigonometric relationships in this figure, we can generalize Eqs. 5-6 to obtain the transformation equations for rotation of a point about any specified mutation position (x_r, y_r):

Fig 5: Rotating a point from position (x,y) to position (x' , y') through an angle θ about rotation point (x_r, y_r).



$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$

$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta \qquad (5\text{-}9)$$

These general rotation equations differ from Eqs. 5-6 by the inclusion of additive terms, as well as the multiplicative factors on the coordinate values. Thus, the matrix expression 5-7 could be modified to include pivot coordinates by matrix addition of a column vector whose elements contain the additive (translational) terms In Eqs. 5-9.

Scaling

A scaling transformation alters the size of an object. This operation can be carried out for polygons by multiplying the coordinate values (x, y) of each vertex by scaling factors s, and s, to produce the transformed coordinates (x', y'):

$$x' = x \cdot s_x, \qquad y' = y \cdot s_y \qquad (5\text{-}10)$$

Scaling factor s, scales objects in the **x** direction, while s_y scales in the x direction. The transformation equations 5-10 can also be written in the matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \qquad (5\text{-}11)$$

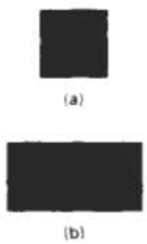
Or

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P} \qquad (5\text{-}12)$$

Where S is the 2 by 2 scaling matrix in Eq. 5-11.

Any positive numeric values can be assigned to the scaling factors s_x and s_y. Values less than 1 reduce the size of objects; values greater than 1 produce an enlargement. Specifying a value of 1 for both s, and s, leaves the size of objects unchanged When s_x and s_y are assigned the same value; a uniform scaling is produced that maintains relative object proportions. Unequal values for s_x and s_y result in a differential scaling that is often used in design applications, whew pictures are constructed from a few basic shapes that can be adjusted by scaling and positioning transformations (Fig.6**).**

Fig 6: Turning a square(a) into rectangle (b) with scaling factor $s_x = 2$ and $s_y = 1$.



Matrix representation of homogeneous coordinate

Many graphics applications involve sequences of geometric transformations. An animation, for example, might require an object to be translated and rotated at each increment of the motion. In design and picture construction applications, we perform translations, rotations, and scaling to tit the picture components into their proper positions. Here we consider how the matrix representations discussed in the previous sections can be reformulated, so that such transformation sequences can be efficiently processed. The term homogeneous coordinate used in mathematics to refer to the effect of this representation on Cartesian equations when a Cartesian point ***(****x*, y) is converted to a homogeneous representation (x_h, y_h, h), equations containing x and y such as

$f(x, y) = 0,$ become homogeneous equations in the three parameters $x_h, y_h,$ and h. 'This just means that if each of that three parameters is replaced by any value ***n*** times that parameter.

Coordinates are represented with three-element column vectors, and transformation operations are written as 3 bv 3 matrices. For translation. we have

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (5\text{-}17)$$

which we can write in the abbreviated form

$$\mathbf{P}' = \mathbf{T}(t_x, t_y) \cdot \mathbf{P} \qquad (5\text{-}18)$$

With $\mathbf{T}(t_x, t_y)$ as the 3 by 3 translation matrix in Eq. 5-17. The inverse of the translation matrix is obtained by replacing the translation parameters t_x and t_y, with their negatives: - $-t_x$ and $-t_y$.

Similarly, rotation transformation equations about the coordinate origin are now written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (5\text{-}19)$$

Or

$$\mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P} \qquad (5\text{-}20)$$

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

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (5\text{-}21)$$

Or

$$P' = S(s_x, s_y) \cdot P \qquad (5\text{-}22)$$

3D GEOMETRIC TRANSFORMATION:

Methods for geometric transformations and object modeling in three dimensions are extended from two-dimensional methods by including considerations for the ***z*** coordinate. We now translate an object by specifying a three-dimensional translation vector, which determines how much the object is to be moved in each of the three coordinate directions. Similarly, we scale an object with three coordinate scaling factors. The extension for three-dimensional rotation is less straightforward. When we discussed two-dimensional rotations in the *xy* plane, we needed to consider only rotations about axes that were perpendicular to the *xy* plane. In three-dimensional space, we can now select any spatial orientation for the rotation axis. Most graphics packages handle three-dimensional rotation as a composite of three rotations, one for each of the three Cartesian axes. Alternatively, a user can easily set up a general rotation matrix, given the orientation of the axis and the q u i r e d rotation angle. As in the two-dimensional case, we express geometric transformations in matrix form. Any sequence of transformations is then represented as, a single matrix, formed by concatenating the matrices for the individual transformations in the sequence.

Translation

In a three-dimensional homogeneous coordinate representation, a point is translated (Fig.1) from position P = *(x*, y, *z)* to position P' = *(x'*, y', z') with the matrix operation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (11\text{-}1)$$

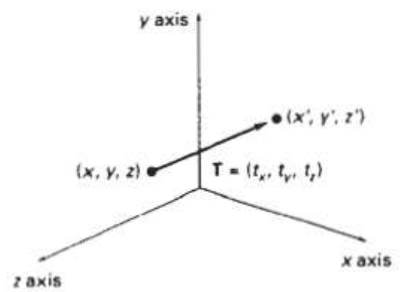
Or

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P} \qquad (11\text{-}2)$$

Parameters $t_x, t_y, and\ t_z$, specifying translation distances for the coordinate directions **x, y,** and z, are assigned any real values. The matrix representation in Eq. 11-1 is equivalent to the three equations,

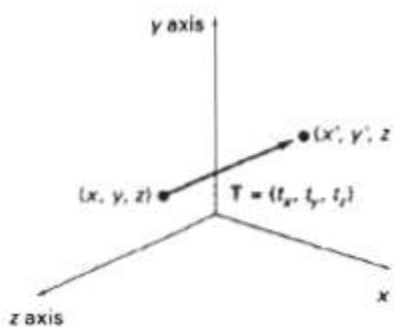
$$x' = x + t_x, \qquad y' = y + t_y, \qquad z' = z + t_z \qquad (11\text{-}3)$$

Fig1: Translating a point with translation vector $\mathbf{T} = (t_x, t_y, t_z)$.



An object is translated in three dimensions by transforming each of the defining points of the object. For an object represented as a set of polygon surfaces, we translate each vertex of each surface (Fig. 2) and redraw the polygon facets in the new position.

Fig: 2 Translating an object with translation vector T.



We obtain the inverse of the translation matrix in Eq. 11-1 by negating the translation distances t,, t,, and t,. This produces a translation in the opposite direction, and the product of a translation matrix and its inverse produces the identity Matrix.

Rotation

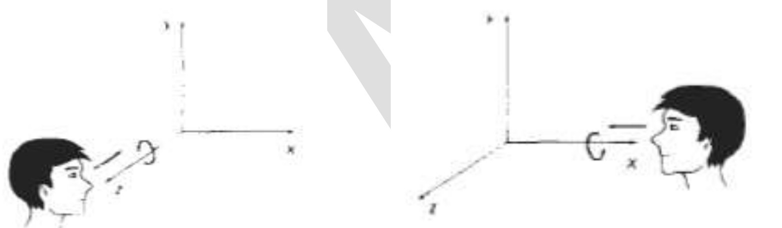
To generate a rotation transformation for an object, we must designate an axis of rotation (about which the object is to be rotated) and the amount of angular rotation. Unlike two- dimensional applications, where all transformations are carried out in the xy plane, a three-dimensional rotation can be specified around any line in space. The easiest rotation axes to handle are those that are parallel to the coordinate axes. Also, we can use combinations of coordinate axis rotations (along with appropriate translations) to specify any general rotation.

By convention, positive rotation angles produce counterclockwise rotations about a coordinate axis, if we are looking along the positive half of the axis toward the coordinate origin (Fig.3). This agrees with our earlier discussion of

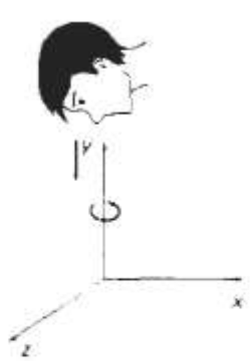
Rotation in two dimensions, where positive rotations in the xy plane are counterclockwise about axes parallel to the z axis.

Fig3: Positive rotation directions about the coordinate axes are

Counterclockwise, when looking toward the origin from a positive coordinate position on each axis.



(a) (b)



(c)

Coordinate-Axes Rotations

The two-dimensional z-axis rotation equations are easily extended to three dimensions:

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta \qquad (11\text{-}4)$$
$$z' = z$$

Parameter θ specifies the rotation angle. In homogeneous coordinate form, the three-dimensional z-axis rotation equations are expressed as,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (11\text{-}5)$$

which we can write more compactly as,

$$\mathbf{P}' = \mathbf{R}_z(\theta) \cdot \mathbf{P} \qquad (11\text{-}6)$$

Figure 4, illustrates rotation of an object about the z axis.

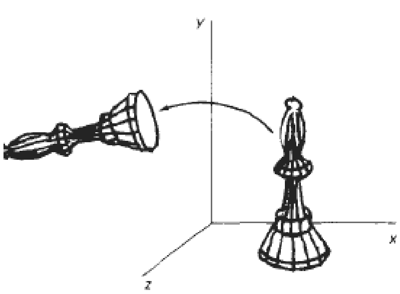


Fig: 4 Rotation of an object about the z axis

Transformation equations for rotations about the other two coordinate axes can be obtained with a cyclic permutation of the coordinate parameters ***x,*** y, and z in Eqs. 11-4. That is, we use the replacements

$$x \rightarrow y \rightarrow z \rightarrow x \qquad (11\text{-}7)$$

as illustrated in Fig.5.

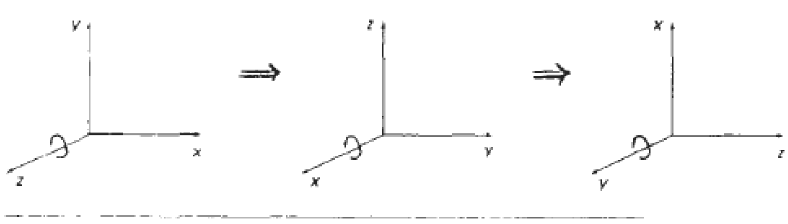
Substituting permutations 11-7 in Eq. 11-4, we get the equations for an x-axis rotation:

$$\begin{aligned} y' &= y\cos\theta - z\sin\theta \\ z' &= y\sin\theta + z\cos\theta \qquad (11\text{-}8) \\ x' &= x \end{aligned}$$

which can be written in the homogeneous coordinate form

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (11\text{-}9)$$

Fig5: Cyclic permutation of the Cartesian-coordinate axes to produce the three sets of coordinate-axis rotation equations.



Cyclically permuting coordinates in Eq. 11-8 give us the transformation equations for a y-axis rotation:

$$z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta \qquad (11\text{-}11)$$
$$y' = y$$

The matrix representation for y-axis rotation is

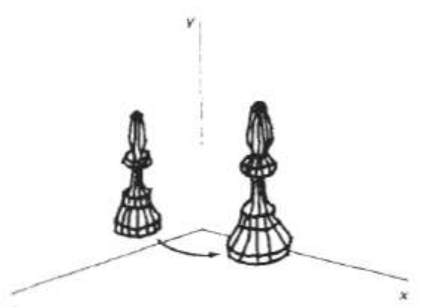
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (11\text{-}12)$$

Or

$$\mathbf{P}' = \mathbf{R}_y(\theta) \cdot \mathbf{P} \qquad (11\text{-}13)$$

An example of y-axis rotation is shown in Fig.5.

Fig: 5 Rotation of an object about the y axis.



An inverse rotation matrix is formed by replacing the rotation angle 0 by $-\theta$. Negative values for rotation angles generate rotations in a clockwise direction, so the identity matrix is produced when any rotation matrix is multiplied by its inverse. Since only the sine function is affected by the change in sign of the rotation angle, the inverse matrix can also be obtained by interchanging rows and columns. That is, we can calculate the inverse of any rotation matrix R by evaluating its transpose $(\mathbf{R}^{-1} = \mathbf{R}^T)$. This method for obtaining an inverse matrix holds also for any composite rotation matrix.

General Three-Dimensional Rotations

A rotation matrix for any axis that does not coincide with a coordinate axis can be set up as a composite transformation involving combinations of translations and the coordinate-axes rotations. We obtain the required composite matrix by first setting up the transformation sequence that moves the selected rotation axis onto one of the coordinate axes. Then we set up the rotation matrix about that coordinate axis for the specified rotation angle. The last step is to obtain the inverse transformation sequence that returns the rotation axis to its original position.

In the special case where an object is to be rotated about an axis that is parallel to one of the coordinate axes, we can attain that

desired rotation with the following transformation sequence.

1. Translate the object so that the rotation axis coincides with the parallel coordinate axis.

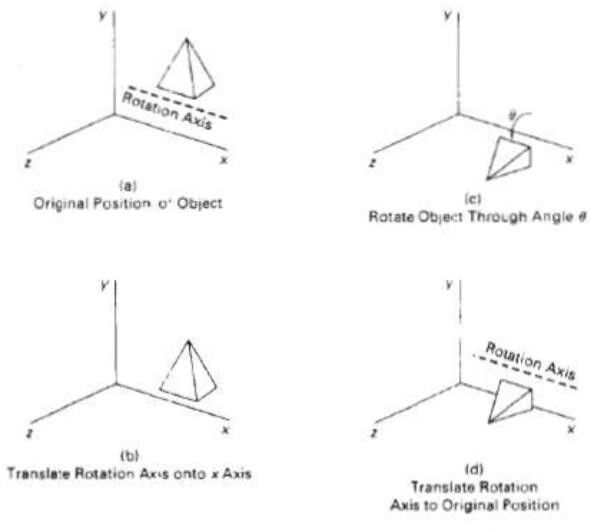
2. Perform the specified rotation about that axis.

3. Translate the object so that the rotation axis is moved back to its original position.

The steps in this sequence are illustrated in Fig. 6. Any coordinate position P on the object in this figure is transformed with the sequence shown as,

$$\mathbf{P}' = \mathbf{T}^{-1} \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T} \cdot \mathbf{P}$$

Fig 6: Sequence of transformations for rotating an object about an axis that is parallel to the *x* axis



where the composite matrix for the transformation is,

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T}$$

which is of the same form as the two-dimensional transformation sequence for rotation about an arbitrary pivot point.

When an object is to be rotated about an axis that **is** not parallel to one of the coordinate axes, we need to perform some additional transformations. In this case, w also need rotations lo align the axis with a selected coordinate axis and to bring the axis hack to its original orientation. Given the specifications for the rotation axis and the rotation angle, we can accomplish the required rotation in five steps.

1 Translate the object so that the rotation axis pass= through the coordinate origin.

2. Rotate the object so that the axis of rotation coincides with one of the coordinate axes.

3. Perform the specified rotation about that coordinate axis.

4. Apply inverse rotations to bring the rotation axis back to its

original orientation.

5. Apply the inverse translation to bring the rotahon axis back to its original position.

Scaling

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

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (11\text{-}42)$$

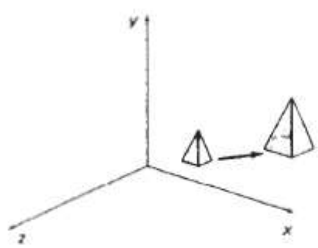
Or

$$\mathbf{P'} = \mathbf{S} \cdot \mathbf{P} \qquad (11\text{-}43)$$

Where scaling parameters s_x, s_y, and s_z are assigned any positive values. Explicit expressions for the coordinate transformations for scaling relative to the origin are

$$x' = x \cdot s_x, \qquad y' = y \cdot s_y, \qquad z' = z \cdot s_z \qquad (11\text{-}44)$$

Fig7: Doubling the sue of an object with transformation 11-42 also moves the object farther from the origin



Scaling an object with transformation 11-42 changes the size of the object and repositions the object relative to the coordinate origin. Also, if the transformation parameters are not all equal, relative dimensions in the object are changed: We preserve the original shape of an object with a uniform scaling $(s_x = s_y = s_z)$. The result of scaling an object uniformly with each scaling parameter set to 2 is shown in Fig.7.

Scaling with respect to a selected fixed position (x_f, y_f, z_f) can be represented with the following transformation sequence:

1. Translate the fixed point to the origin.

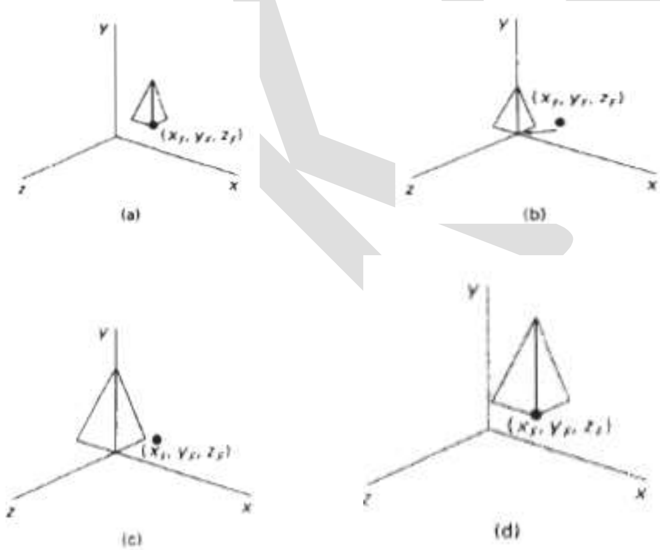
2. Scale the object relative to the coordinate origin using Eq. 11-42.

3. Translate the fixed point back to its original position.

This sequence of transformations is demonstrated in Fig. 8. The matrix representation for an arbitrary fixed-point scaling can then be expressed as the concatenation of these translate-scale-translate transformations as

$$\mathbf{T}(x_f, y_f, z_f) \cdot \mathbf{S}(s_x, s_y, s_z) \cdot \mathbf{T}(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11\text{-}45)$$

Fig8: Scaling an object relative to the selected fixed point is equivalent to the sequence of transformations shown.



We form the inverse scaling matrix for either Eq. 11-42 or Eq. 11-45 by replacing the scaling parameters s_x, s_y, and s_z with their reciprocals. The inverse matrix generates an opposite scaling transformation, so the concatenation of any scaling matrix and its inverse produces the identity matrix.

VANISHING POINTS:

Certain set of parallel lines appear to meet at a different point – The Vanishing point for this direction

• Principal vanishing points are formed by the apparent intersection of lines parallel to one of the three principal x, y, z axes.

• The number of principal vanishing points is determined by the number of principal axes intersected by the view plane.

• Sets of parallel lines on the same plane lead to collinear vanishing points. – The line is called the horizon for that plane

Classes of Perspective Projection • One-Point Perspective • Two-Point Perspective • Three-Point Perspective

POSSIBLE QUESTIONS

2 MARKS

1. Distinguish between Scaling and Rotation in 2D Geometric transformation.
2. Explain the methods of line drawing algorithm
3. What is vanishing points?

POSSIBLE QUESTIONS

6 MARKS

1. Explain in detail about the two dimensional viewing transformation pipeline.
2. Explain in detail about Circle and ellipse drawing algorithm
3. Describe line and polygon clipping algorithm in detail
4. Explain in detail about the three dimensional transformation with neat sketch.

UNIT-IV

SYLLABUS

Geometric Modeling : Representing curves & Surfaces.

GEOMETRIC MODELLING

CURVES:

Types of Curves

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

Implicit Curves

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

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

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

$$\mathbf{x^2 + y^2 - R^2 = 0}$$

Explicit Curves

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

Parametric Curves

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

P(t) = f(t), g(t) or P(t) = x(t), y(t)

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

Bezier Curves

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

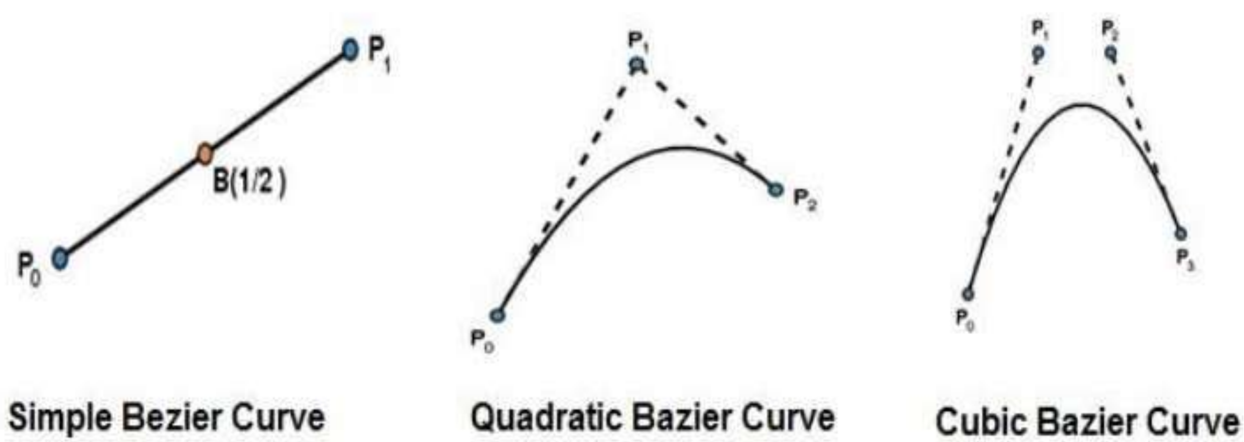
∑k=0nPiBni(t)∑k=0nPiBin(t)

Where pipi is the set of points and Bni(t)Bin(t) represents the Bernstein polynomials which are given by −

Bni(t)=(ni)(1−t)n−itiBin(t)=(ni)(1−t)n−iti

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

The simplest Bézier curve is the straight line from the point P0P0 to P1P1. A quadratic Bezier curve is determined by three control points. A cubic Bezier curve is determined by four control points.



Properties of Bezier Curves

Bezier curves have the following properties −

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.

- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

B-Spline Curves

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

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

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

A B-spline curve is defined as a linear combination of control points Pi and B-spline basis function Ni,Ni, k (t) given by

C(t)=∑ni=0PiNi,k(t),C(t)=∑i=0nPiNi,k(t), n≥k−1,n≥k−1, tϵ[tk−1,tn+1]tϵ[tk−1,tn+1]

Where,

- {pipi: i=0, 1, 2….n} are the control points

- k is the order of the polynomial segments of the B-spline curve. Order k means that the curve is made up of piecewise polynomial segments of degree k - 1,
- the Ni,k(t)Ni,k(t) are the “normalized B-spline blending functions”. They are described by the order k and by a non-decreasing sequence of real numbers normally called the “knot sequence”.

ti:i=0,...n+Kti:i=0,...n+K

The N_i, k functions are described as follows −

Ni,1(t)={1,0,ifuϵ[ti,ti+1)OtherwiseNi,1(t)={1,ifuϵ[ti,ti+1)0,Otherwise

and if k > 1,

Ni,k(t)=t−titi+k−1Ni,k−1(t)+ti+k−tti+k−ti+1Ni+1,k−1(t)Ni,k(t)=t−titi+k−1Ni,k−1(t)+ti+k−tti+k−ti+1Ni+1,k−1(t)

and

tϵ[tk−1,tn+1)tϵ[tk−1,tn+1)

Properties of B-spline Curve

B-spline curves have the following properties −

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

- The curve line within the convex hull of its defining polygon.

SURFACES:

Polygon Surfaces

Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

- **Boundary Representations (B-reps)** − It describes a 3D object as a set of surfaces that separates the object interior from the environment.
- **Space–partitioning representations** − It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids (usually cubes).

The most commonly used boundary representation for a 3D graphics object is a set of surface polygons that enclose the object interior. Many graphics system use this method. Set of polygons are stored for object description. This simplifies and speeds up the surface rendering and display of object since all surfaces can be described with linear equations.

The polygon surfaces are common in design and solid-modeling applications, since their **wireframe display** can be done quickly to give general indication of surface structure. Then realistic scenes are produced by interpolating shading patterns across polygon surface to illuminate.

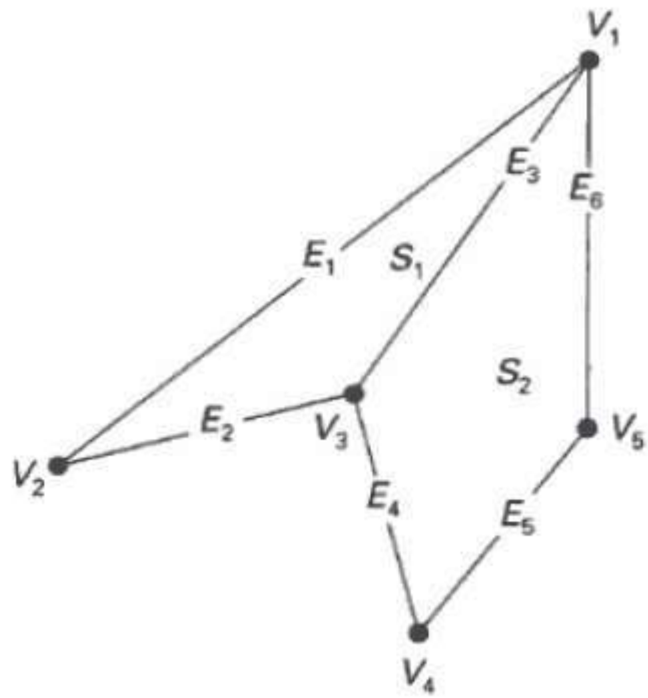
A 3D object represented by polygons

Polygon Tables

In this method, the surface is specified by the set of vertex coordinates and associated attributes. As shown in the following figure, there are five vertices, from v_1 to v_5.

- Each vertex stores x, y, and z coordinate information which is represented in the table as v_1: x_1, y_1, z_1.

- The Edge table is used to store the edge information of polygon. In the following figure, edge E_1 lies between vertex v_1 and v_2 which is represented in the table as E_1: v_1, v_2.
- Polygon surface table stores the number of surfaces present in the polygon. From the following figure, surface S_1 is covered by edges E_1, E_2 and E_3 which can be represented in the polygon surface table as S_1: E_1, E_2, and E_3.



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

Plane Equations

The equation for plane surface can be expressed as −

$$Ax + By + Cz + D = 0$$

Where (x, y, z) is any point on the plane, and the coefficients A, B, C, and D are constants describing the spatial properties of the plane. We can obtain the values of A, B, C, and D by solving a set of three plane equations using the coordinate values for three non collinear points in the plane. Let us assume that three vertices of the plane are (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3).

Let us solve the following simultaneous equations for ratios A/D, B/D, and C/D. You get the values of A, B, C, and D.

$$(A/D)\, x_1 + (B/D)\, y_1 + (C/D)\, z_1 = -1$$

$$(A/D)\, x_2 + (B/D)\, y_2 + (C/D)\, z_2 = -1$$

$$(A/D)\, x_3 + (B/D)\, y_3 + (C/D)\, z_3 = -1$$

To obtain the above equations in determinant form, apply Cramer's rule to the above equations.

$A = \begin{bmatrix} 1& y_{1}& z_{1}\\ 1& y_{2}& z_{2}\\ 1& y_{3}& z_{3} \end{bmatrix} B = \begin{bmatrix} x_{1}& 1& z_{1}\\ x_{2}& 1& z_{2}\\ x_{3}& 1& z_{3} \end{bmatrix} C = \begin{bmatrix} x_{1}& y_{1}& 1\\ x_{2}& y_{2}& 1\\ x_{3}& y_{3}& 1 \end{bmatrix} D = - \begin{bmatrix} x_{1}& y_{1}& z_{1}\\ x_{2}& y_{2}& z_{2}\\ x_{3}& y_{3}& z_{3} \end{bmatrix}$

For any point (x, y, z) with parameters A, B, C, and D, we can say that −

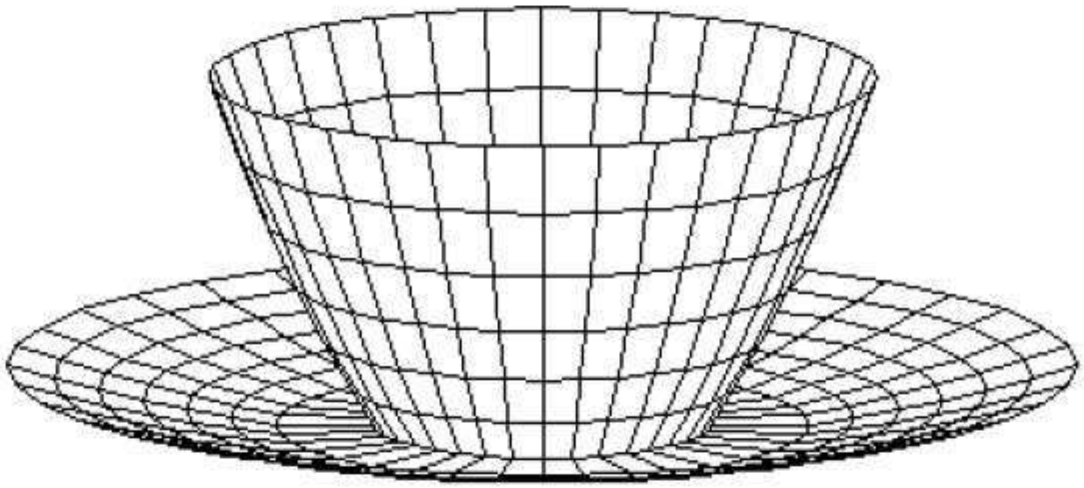
- $Ax + By + Cz + D \neq 0$ means the point is not on the plane.
- $Ax + By + Cz + D < 0$ means the point is inside the surface.
- $Ax + By + Cz + D > 0$ means the point is outside the surface.

Polygon Meshes

3D surfaces and solids can be approximated by a set of polygonal and line elements. Such surfaces are called **polygonal meshes**. In polygon mesh, each edge is shared by at most two polygons. The set of polygons or faces, together form the "skin" of the object.

This method can be used to represent a broad class of solids/surfaces in graphics. A polygonal mesh can be rendered using hidden surface removal algorithms. The polygon mesh can be represented by three ways −

- Explicit representation
- Pointers to a vertex list
- Pointers to an edge list



Advantages

- It can be used to model almost any object.
- They are easy to represent as a collection of vertices.
- They are easy to transform.
- They are easy to draw on computer screen.

Disadvantages

- Curved surfaces can only be approximately described.
- It is difficult to simulate some type of objects like hair or liquid.

POSSIBLE QUESTIONS

2 MARKS

1. What is geometrics modeling?
2. Define representing curves?
3. What is surfaces in computer graphics?

8 MARKS

1. Explain in detail about representing curves with neat sketch.
2. Describe about surface sin computer graphics with neat diagram.

UNIT-V

SYLLABUS

Visible Surface determination : Hidden surface elimination. **Surface rendering :** Illumination and shading models. Basic color models and Computer Animation.

VISIBLE SURFACE DETERMINATION

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

There are two approaches for removing hidden surface problems − **Object-Space method** and **Image-space method**. The Object-space method is implemented in physical coordinate system and image-space method is implemented in screen coordinate system.

When we want to display a 3D object on a 2D screen, we need to identify those parts of a screen that are visible from a chosen viewing position.

Depth Buffer (Z-Buffer) Method

This method is developed by Cutmull. It is an image-space approach. The basic idea is to test the Z-depth of each surface to determine the closest (visible) surface.

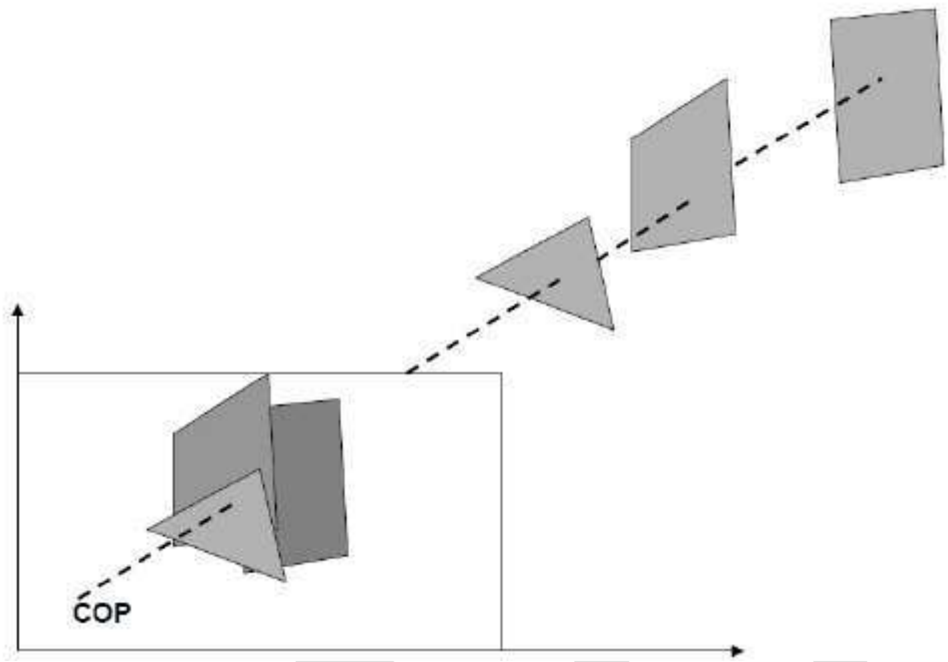
In this method each surface is processed separately one pixel position at a time across the surface. The depth values for a pixel are compared and the closest (smallest z) surface determines the color to be displayed in the frame buffer.

It is applied very efficiently on surfaces of polygon. Surfaces can be processed in any order. To override the closer polygons from the far ones, two buffers named **frame buffer** and **depth buffer,** are used.

Depth buffer is used to store depth values for (x, y) position, as surfaces are processed ($0 \leq depth \leq 1$).

The **frame buffer** is used to store the intensity value of color value at each position (x, y).

The z-coordinates are usually normalized to the range [0, 1]. The 0 value for z-coordinate indicates back clipping pane and 1 value for z-coordinates indicates front clipping pane.



Algorithm

Step-1 − Set the buffer values −

Depthbuffer (x, y) = 0

Framebuffer (x, y) = background color

Step-2 − Process each polygon (One at a time)

For each projected (x, y) pixel position of a polygon, calculate depth z.

If Z > depthbuffer (x, y)

Compute surface color,

set depthbuffer (x, y) = z,

framebuffer (x, y) = surfacecolor (x, y)

Advantages

- It is easy to implement.
- It reduces the speed problem if implemented in hardware.

- It processes one object at a time.

Disadvantages

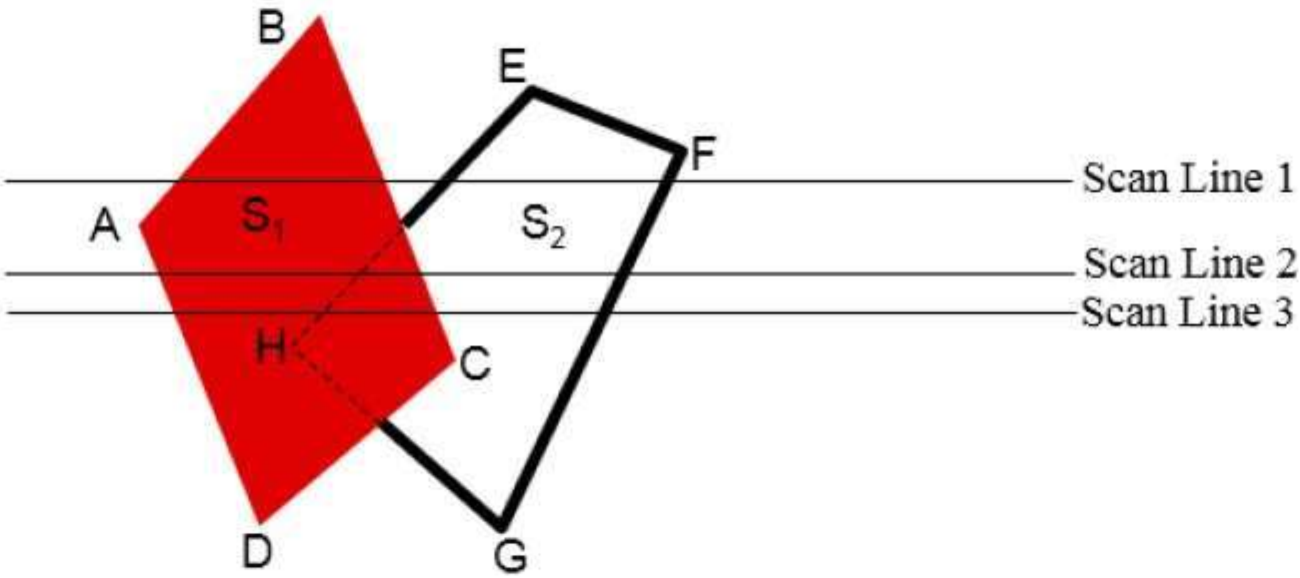
- It requires large memory.
- It is time consuming process.

Scan-Line Method

It is an image-space method to identify visible surface. This method has a depth information for only single scan-line. In order to require one scan-line of depth values, we must group and process all polygons intersecting a given scan-line at the same time before processing the next scan-line. Two important tables, **edge table** and **polygon table,** are maintained for this.

The Edge Table − It contains coordinate endpoints of each line in the scene, the inverse slope of each line, and pointers into the polygon table to connect edges to surfaces.

The Polygon Table − It contains the plane coefficients, surface material properties, other surface data, and may be pointers to the edge table.



To facilitate the search for surfaces crossing a given scan-line, an active list of edges is formed. The active list stores only those edges that cross the scan-line in order of increasing x. Also a flag is set for each surface to indicate whether a position along a scan-line is either inside or outside the surface.

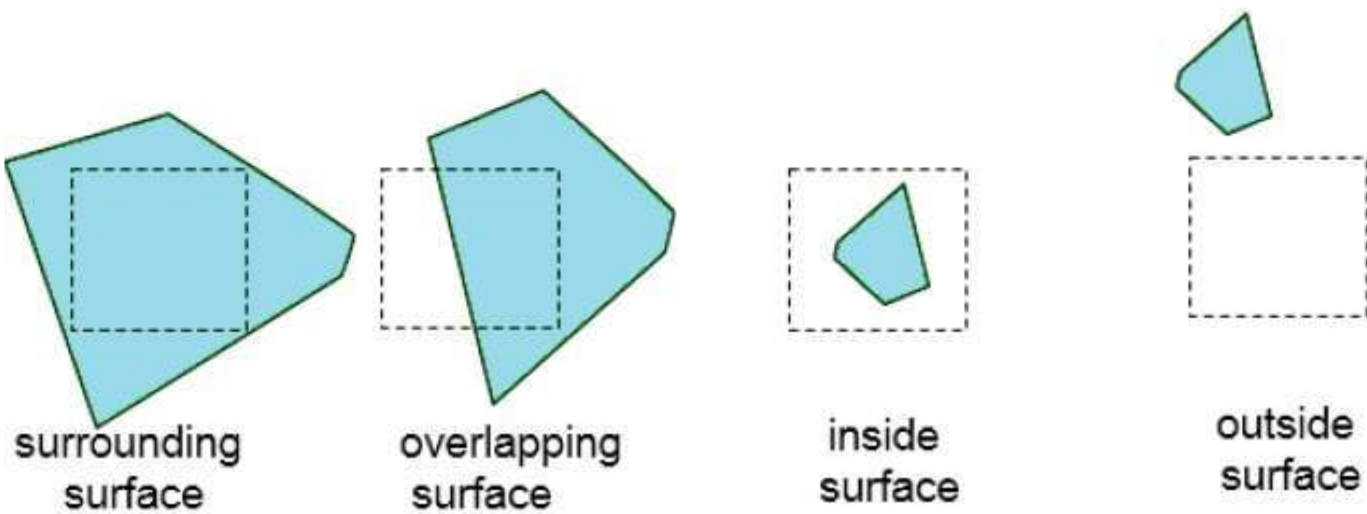
Pixel positions across each scan-line are processed from left to right. At the left intersection with a surface, the surface flag is turned on and at the right, the flag is turned off. You only need to perform depth calculations when multiple surfaces have their flags turned on at a certain scan-line position.

Area-Subdivision Method

The area-subdivision method takes advantage by locating those view areas that represent part of a single surface. Divide the total viewing area into smaller and smaller rectangles until each small area is the projection of part of a single visible surface or no surface at all.

Continue this process until the subdivisions are easily analyzed as belonging to a single surface or until they are reduced to the size of a single pixel. An easy way to do this is to successively divide the area into four equal parts at each step. There are four possible relationships that a surface can have with a specified area boundary.

- **Surrounding surface** − One that completely encloses the area.
- **Overlapping surface** − One that is partly inside and partly outside the area.
- **Inside surface** − One that is completely inside the area.
- **Outside surface** − One that is completely outside the area.



The tests for determining surface visibility within an area can be stated in terms of these four classifications. No further subdivisions of a specified area are needed if one of the following conditions is true −

- All surfaces are outside surfaces with respect to the area.
- Only one inside, overlapping or surrounding surface is in the area.

- A surrounding surface obscures all other surfaces within the area boundaries.

Back-Face Detection

A fast and simple object-space method for identifying the back faces of a polyhedron is based on the "inside-outside" tests. A point (x, y, z) is "inside" a polygon surface with plane parameters A, B, C, and D if When an inside point is along the line of sight to the surface, the polygon must be a back face (we are inside that face and cannot see the front of it from our viewing position).

We can simplify this test by considering the normal vector **N** to a polygon surface, which has Cartesian components (A, B, C).

In general, if V is a vector in the viewing direction from the eye (or "camera") position, then this polygon is a back face if

V.N > 0

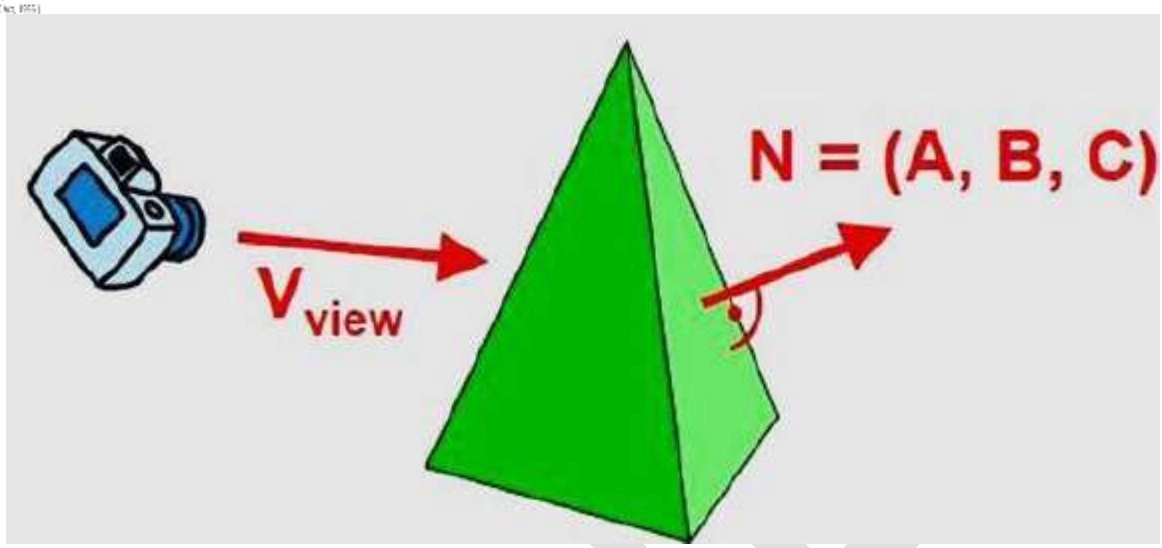
Furthermore, if object descriptions are converted to projection coordinates and your viewing direction is parallel to the viewing z-axis, then −

$V = (0, 0, V_z)$ and $V.N = V_Z C$

So that we only need to consider the sign of C the component of the normal vector **N**.

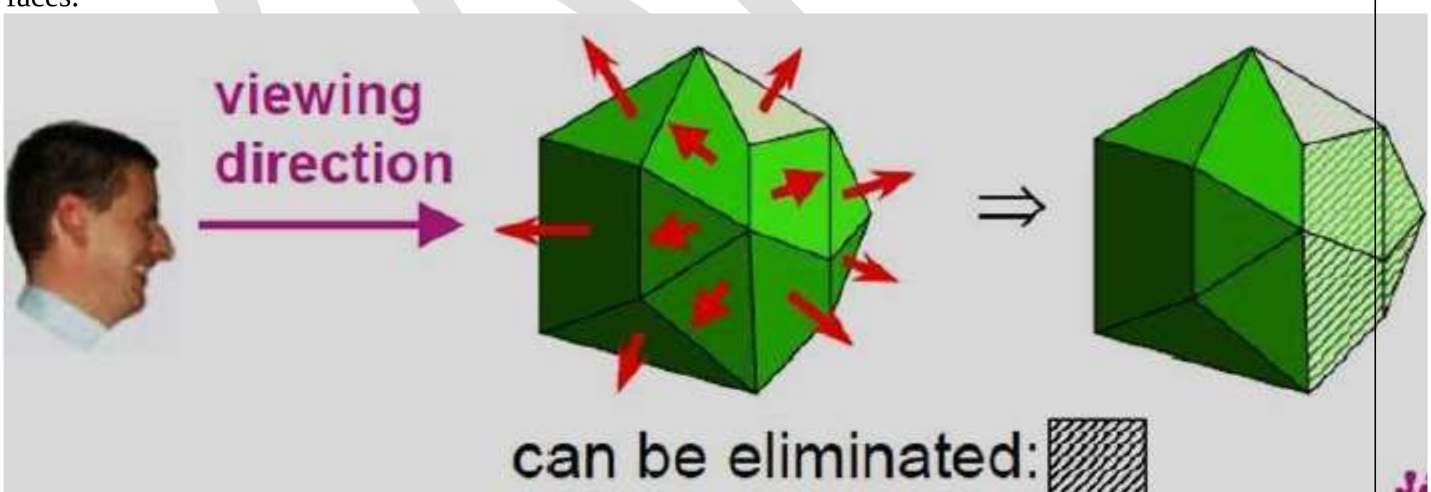
In a right-handed viewing system with viewing direction along the negative ZVZV axis, the polygon is a back face if C < 0. Also, we cannot see any face whose normal has z component C = 0, since your viewing direction is towards that polygon. Thus, in general, we can label any polygon as a back face if its normal vector has a z component value −

C <= 0



Similar methods can be used in packages that employ a left-handed viewing system. In these packages, plane parameters A, B, C and D can be calculated from polygon vertex coordinates specified in a clockwise direction (unlike the counterclockwise direction used in a right-handed system).

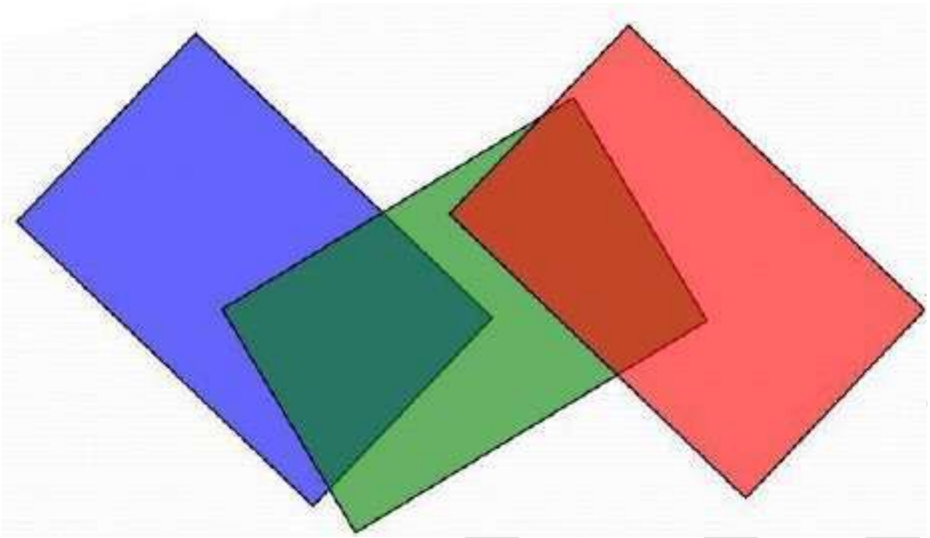
Also, back faces have normal vectors that point away from the viewing position and are identified by C >= 0 when the viewing direction is along the positive ZvZv axis. By examining parameter C for the different planes defining an object, we can immediately identify all the back faces.



A-Buffer Method

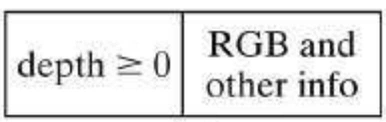
The A-buffer method is an extension of the depth-buffer method. The A-buffer method is a visibility detection method developed at Lucas film Studios for the rendering system Renders Everything You Ever Saw (REYES).

The A-buffer expands on the depth buffer method to allow transparencies. The key data structure in the A-buffer is the accumulation buffer.

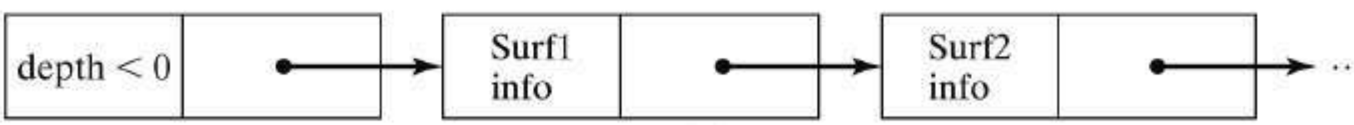


Each position in the A-buffer has two fields −

- **Depth field** − It stores a positive or negative real number
- **Intensity field** − It stores surface-intensity information or a pointer value



(a)



(b)

If depth >= 0, the number stored at that position is the depth of a single surface overlapping the corresponding pixel area. The intensity field then stores the RGB components of the surface color at that point and the percent of pixel coverage.

If depth < 0, it indicates multiple-surface contributions to the pixel intensity. The intensity field then stores a pointer to a linked list of surface data. The surface buffer in the A-buffer includes −

- RGB intensity components
- Opacity Parameter
- Depth
- Percent of area coverage
- Surface identifier

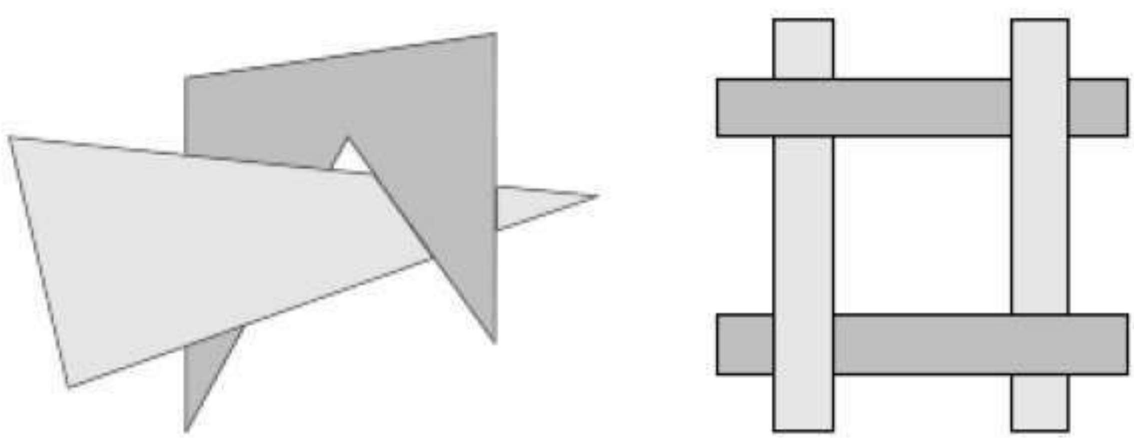
The algorithm proceeds just like the depth buffer algorithm. The depth and opacity values are used to determine the final color of a pixel.

Depth Sorting Method

Depth sorting method uses both image space and object-space operations. The depth-sorting method performs two basic functions −

- First, the surfaces are sorted in order of decreasing depth.
- Second, the surfaces are scan-converted in order, starting with the surface of greatest depth.

The scan conversion of the polygon surfaces is performed in image space. This method for solving the hidden-surface problem is often referred to as the **painter's algorithm**. The following figure shows the effect of depth sorting −



The algorithm begins by sorting by depth. For example, the initial “depth” estimate of a polygon may be taken to be the closest z value of any vertex of the polygon.

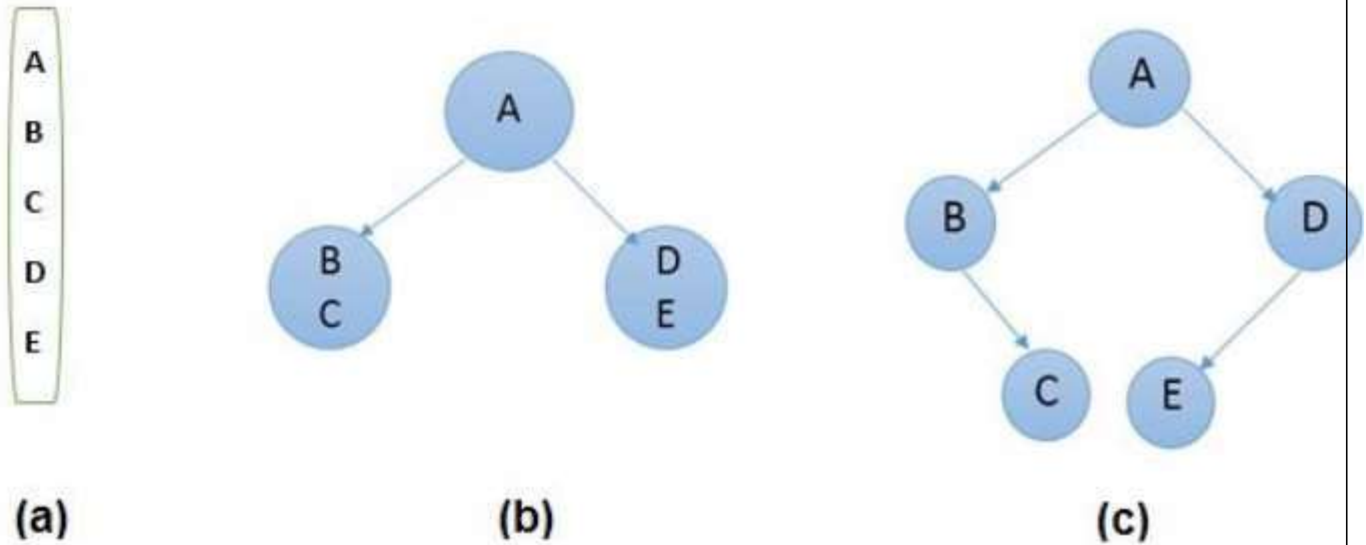
Let us take the polygon P at the end of the list. Consider all polygons Q whose z-extents overlap P’s. Before drawing P, we make the following tests. If any of the following tests is positive, then we can assume P can be drawn before Q.

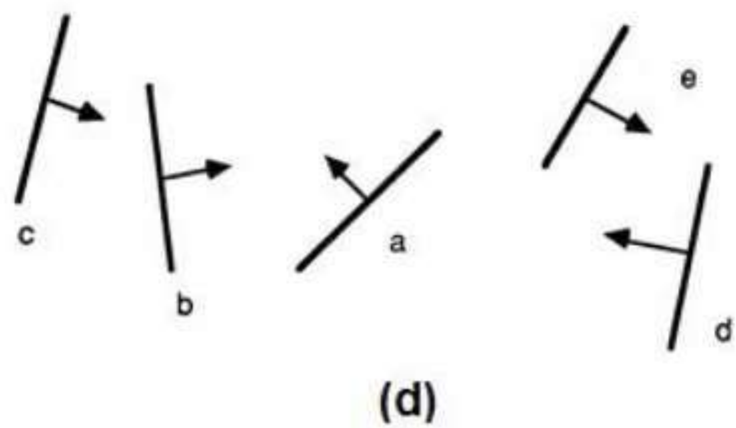
- Do the x-extents not overlap?
- Do the y-extents not overlap?
- Is P entirely on the opposite side of Q's plane from the viewpoint?
- Is Q entirely on the same side of P's plane as the viewpoint?
- Do the projections of the polygons not overlap?

If all the tests fail, then we split either P or Q using the plane of the other. The new cut polygons are inserting into the depth order and the process continues. Theoretically, this partitioning could generate $O(n^2)$ individual polygons, but in practice, the number of polygons is much smaller.

Binary Space Partition (BSP) Trees

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





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

SURFACE RENDERING :

Illumination and shading models:

Illumination Models

• Motivation: In order to produce realistic images, we must simulate the appearance of surfaces under various lighting conditions

• Illumination Model: Given the illumination incident at a point on a surface, quantifies the reflected light

Lighting effects are described with models that consider the interaction of light sources with object surfaces

• The factors determining the lighting effects are: – The light source parameters:

• Positions

• Electromagnetic Spectrum

• Shape – The surface parameters

• Position

• Reflectance properties

• Position of nearby surfaces

The eye (camera) parameters

• Position

• Sensor spectrum sensitivities

An illumination model is used to calculate the intensity of the light that is reflected at a given point on a surface

• A rendering method uses intensity calculations from the illumination model to determine the light intensity at all pixels in the image

Light Source Models

• Point Source (a): All light rays originate at a point and radially diverge. A reasonable approximation for sources whose dimensions are small compared to the object size

• Parallel source (b): Light rays are all parallel. May be modeled as a point source at infinite distance (the sun)

• Distributed source (c): All light rays originate at a finite area in space. It models a nearby source, such as a fluorescent light

Simplified and fast methods for calculating surfaces intensities, mostly empirical

• Calculations are based on optical properties of surfaces and the lighting conditions (no reflected sources nor shadows)

• Light sources are considered to be point sources

• Reasonably good approximation for most scenes

Assume there is some non-directional light in the environment (background light)

• The amount of ambient light incident on each object is constant for all surfaces and over all directions • Very simple model, not very realistic

• OpenGL default

Flat Shading

• A single intensity is calculated for each surface polygon

• Fast and simple method

• Gives reasonable result only if all of the following assumptions are valid: – The object is a polyhedron – Light source is far away from the surface so that N

•L is constant over each polygon – Viewing position is far away from the surface so that V

•R is constant over each polygon

Renders the polygon surface by linearly interpolating intensity values across the surface Gouraud Shading Algorithm: 1. Determine the normal at each polygon vertex 2. Apply an illumination model to each vertex to calculate the vertex intensity 3. Linearly interpolate the vertex intensities over the surface polygon.

BASIC COLOR MODELS AND COMPUTER ANIMATION:

The RGB Color Model

The red-green-blue model is formed by a color cube $\{(R,G,B) : 0 \leq R,G,B \leq 1\}$.

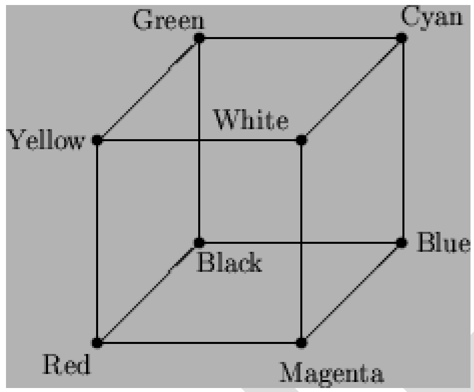


Figure: The RGB-cube

Conversion from (R,G,B) to (X,Y,Z) is given via the chromaticities (X_r, Y_r, Z_r), (X_g, Y_g, Z_g) and (X_b, Y_b, Z_b) of the CRTs phosphors by matrix multiplication via:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} Xr & X_g & X_b \\ Yr & Y_g & Y_b \\ Zr & Z_g & Z_b \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Let $C_r := X_r + Y_r + Z_r$. Then $X_r = x_r \cdot C_r$, $Y_r = y_r \cdot C_r$ and $Z_r = z_r \cdot C_r = (1 - x_r - y_r) \cdot C_r$.

This can be calculated from $X = \frac{x}{y}Y$, $Y = Y$, $Z = \frac{1-x-y}{y}Y$.

The CMY Color Model

This stands for cyan-magenta-yellow and is used for hardcopy devices. In contrast to color on the monitor, the color in printing acts subtractive and not additive. A printed color that looks red absorbs the other two components G and B and reflects R. Thus its (internal) color is G+B=CYAN. Similarly R+B=MAGENTA and R+G=YELLOW. Thus the C-M-Y coordinates are just the complements of the R-G-B coordinates:

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

If we want to print a red looking color (i.e. with R-G-B coordinates (1,0,0)) we have to use C-M-Y values of (0,1,1). Note that M absorbs G, similarly Y absorbs B and hence $M+Y$ absorbs all but R.

Black ($(R,G,B)=(0,0,0)$) corresponds to $(C,M,Y)=(1,1,1)$ which should in principle absorb R, G and B. But in practice this will appear as some dark gray. So in order to be able to produce better contrast printers often use black as 4^{th} color. This is the CMYK-model. Its coordinates are obtained from that of the CMY-model by $K := \max(C,M,Y)$, $C := C - K$, $M := M - K$ and $Y := Y - K$.

The YIQ Color Model

This is used for color TV. Here Y is the luminance (the only component necessary for B&W-TV). The conversion from RGB to YIQ is given by

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

for standard NTSC RGB phosphor with chromaticity values

	R	G	B
x	0.67	0.21	0.14
y	0.33	0.71	0.08

The advantage of this model is that more bandwidth can be assigned to the Y-component (luminance) to which the human eye is more sensible than to color information. So for NTSC TV there are 4MHz assigned to Y, 1.5MHz to I and 0.6MHz to Q.

The HSV color model

All color models treated so far are hardware oriented. The Hue-Saturation-Value model is oriented towards the user/artist. The allowed coordinates fill a six sided pyramid the 3 top faces of the color cube as base. Note that at the same height colors of different perceived brightness are positioned. Value is given by the height, saturation is coded in the distance from the axes and hue by the position on the boundary.

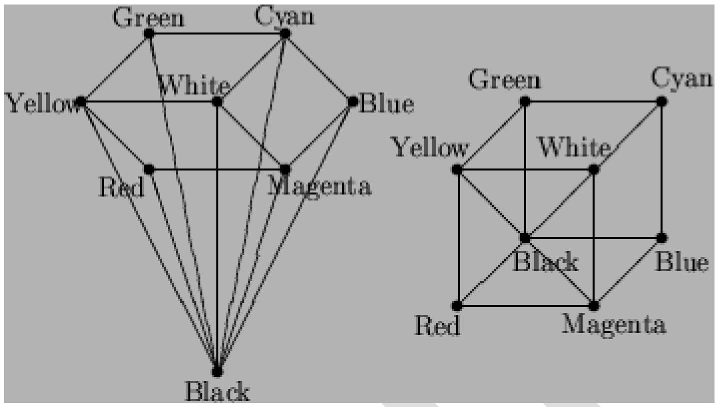


Figure: The HSV-model versus the RGB-model

Note that conversion from RGB to HSV is given by affine coordinate changes on each of the 3 four-sided sub-pyramids corresponding each to 1/3 of the color cube.

The HLS Color Model

Here the RGB-cube is deformed in such a way that a six sided double pyramid results with the same base as in the HSV-model, but with two tips at black and at white.

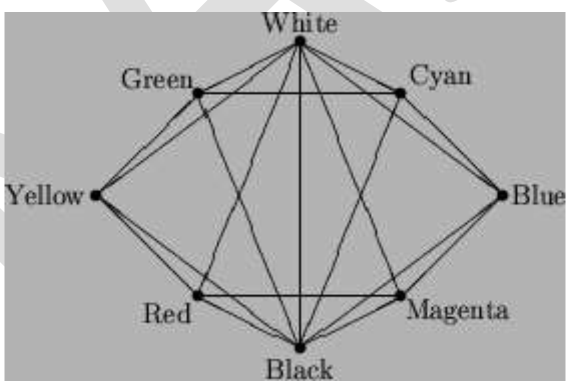


Figure: The HLS-model

POSSIBLE QUESTIONS

2 MARKS

1. What is Visible Surface determination?

2. Define hidden surface elimination.
3. What is rendering?

8 MARKS

1. Explain about visible surface determination in detail.
2. Describe surface rendering in detail with neat sketch.
3. Explain about illumination and shading model.

KARPAGAM ACADEMY OF HIGHER EDUCATION								
DEPARTMENT OF COMPUTER SCIENCE, CA & IT								
II B.Sc CS (Batch 2016-2019)								
COMPUTER GRAPHICS								
PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS								
ONLINE EXAMINATIONS								
UNIT-I								
Sno	**Questions**	**Opt1**	**Opt2**	**Opt3**	**Opt4**	**Opt5**	**Opt6**	**Answer**
1	The ________ ______ display is used in television sets	DVST	storage tube	cathode ray tube	laser scan			cathode ray tube
2	A ________ _________ ___ is a set of subroutines that provide high level access to graphics input output hardware	application	graphics package	hardware	subprogram			graphics package
3	The ratio of image's width to its height, measured in unit length or no of pixels is called_____ _	resolution	raster	size	aspect ratio			aspect ratio
4	Duration of phosphorescence exhibited by the phosphor is called ________	persistance	phophorescence	fluorescence	existence			persistance

5	The smallest picture unit accepted by display is termed as a ________________	dot	pixel	bit	point			pixel
6	A letter or a digits is formed by ________________ of pixels	linear	group	multiple	matrix			matrix
7	________ is a branch of computer science that deals with theory and technology for computerized image synthesis	Animated Graphics	General Graphics	Multimedia Graphics	Computer Graphics			Computer Graphics
8	The area of computer graphics that is responsible for converting a continuous figure into discrete approximation is called ________	scan conversion	transformation	representation	translation			scan conversion
9	The Surface area that are obscured from our eyesight.	Hidden surface	Hidden surface removal	Obscured surface	Removed surface			Hidden surface removal

10	Pixel is	A memory block	An input device	A data structure	Smallest addressable point on the screen			Smallest addressable point on the screen
11	Resolution is defined as	The number of pixels in the horizontal direction + the number of pixels in the vertical direction	The number of pixels in the vertical direction - the number of pixels in the horizontal direction	The number of pixels in the horizontal direction x the number of pixels in the vertical direction	The number of pixels in the vertical direction x the number of pixels in the horizontal direction			The number of pixels in the horizontal direction x the number of pixels in the vertical direction
12	Aspect Ratio is	The ratio of image's height	The ratio of image's width to its height	The ration of image's intensity levels	The ration of window to viewport height			The ratio of image's width to its height
13	Refresh rate is	The frequency at which the aliasing takes place	The rate at which the number of bit planes are accessed at a given time	The rate at which the picture is redrawn	The frequency at which the contents of the frame buffer is sent to the display monitor			The frequency at which the contents of the frame buffer is sent to the display monitor
14	World coordinate system is	The coordinate system in which the image is defined	The coordinate system in which the object is defined	The coordinate system in which the surface is defined	The coordinate system in which the transformation are defined			The coordinate system in which the object is defined
15	Frame buffer is	The device used for displaying the colors of an image	The memory area in which the graphics package is stored	The device which controls the refresh rate	The memory area in which the image, being displayed, is stored			The memory area in which the image, being displayed, is stored
16	Aliasing means	Cueing effect	Cumming effect	Staircase effect	Rendering effect			Staircase effect

17	_______ is an abstract, continuous space which defines the object.	discrete space	continuous space	object space	air space			object space
18	The purpose of the __________ __________ is to convert electrical signals into visible images	display controller	digital device	digital controller	display device			display device
19	The __________ __________ ___ receive information from the computer and convert it into signals acceptable to the device	digital device	display controller	digital controller	display device			display controller
20	The __________ _ convert computer's electrical signals into visible images at high speed	DVST	laser scan	CRT	plasma panel			CRT
21	The __________ ________ in CRT emits on high velocity fine focus	electron	phosphor	yoke	sulphur			electron

22	________ ________ is coated on inside CRT, which glow when the electron beam strikes on it.	yoke	electron gun	sulphur	phosphor			phosphor
23	The ________ ________ in CRT, deflect the electron beam to different parts of tube face when current pass through the coils	electron gun	sulphur	deflection plates	phosphor			deflection plates
24	A steady picture is maintaine d by tracing the electron beam rapidly and repeatedly is called ________ ________ process	refresh	clear	clean	erase			refresh
25	The brightness of the image is controlled by altering the ________ ________ in electron gun	cathode	control grid	filament	anode			control grid

26	Area of Graphics which deals with placement of object in various size, orientation or location is called _________	scan conversion	transformation	representation	translation			transformation
27	The distortion introduced by conversion from continuous space to discrete space is referred to as_________	Aliasing	anti aliasing	clipping	viewing			Aliasing
28	The area of CG that converts a continuous figure to its discrete approximation is called ___	scan conversion	transformation	representation	translation			scan conversion
29	Which of these RGB value represent a grey colour	(0.4,0.5,0.0)	0.2,1.0,0.0)	(1,0,0)	(0.5,0.5,0.5)			(0.5,0.5,0.5)
30	_________ is the complementary color model to RGB	XYZ	BGR	CMY	ABC			CMY

31	The halftone grid patterns are also called ___________	dither patterns	color patterns	image patterns	pixel pattern			dither patterns
32	__________ 24 bit representation provides a wide range of colors (16.7 million colours)	Lookup table	RGB color space	direct coding	CMY color cube			direct coding
33	In colour display the distace between the center of two adjacent dot patterns is called the ________	pitch	intensity	volume	brightness			pitch
34	In ____________, the size of the dots are inversely proportioal to the intended intensity level	printing	halftoning	dithering	error diffusion			dithering
35	__________ protocol is used to return the colour or index value of a pixel	getValue( )	getIndex( )	getColor( )	getPixel( )			getPixel( )

36	The________ is the most popular and successful image presentation device.	Printer	Pixel	CRT	Dither			Printer
37	In Error Diffusion technique the error term is propagated to ___________ for compensation.	all the neighboring pixels	neighboring pixels in the next scan line	yet to be processed neighboring pixels	neighboring pixel at the left			yet to be processed neighboring pixels
38	The digital image is often encoded in the form of __________	binary files	Unicode files	text files	ASCII files			binary files
39	A 512 x 512 raster requires _____ bits in a bit plane	218	212	210	28			218
40	What is dpi	Dots per inch	Digits per inch	Division per inch	None of the above			Dots per inch
41	A gray-scale image is typically coded with _________ bits per pixel	1	4	8	24			8
42	The _____ bit format is commonly referred to as the ________representation	Real color	Bright color	True color	brightness			True color

43	The look up table technique _________ the number of intensity levels	Gives no effect	Removes	Increase	Decrease			Increase
44	A simple 3-bit plane frame buffer can have ____________ number of color combination	4	24	16	8			8
45	Image data format specifies the ________in which pixel values are stored in the image data section	Value	order	Precedence	Successiveness			order
46	TIFF refers to	Tiff-Image file format	Triggered Image File format	Tagged Image File Format	Tilled Image Formatted File			Tagged Image File Format
47	JPEG refers to	Joint Picture Expert Group	Join picture Expert Group	Joint Photographic Experts Group	join picture externel group			Joint Photographic Experts Group
48	RGB refers to	Resolution-Global-Bright	Resolution of Green and Blue	Red-Green-Blue	red resolution blue			Red-Green-Blue
49	Which of the following are image file	.doc	.xls	.http	.jpeg			.jpeg
50	The color black and white occupies	1 bit per pixel	2 bit per pixel	4 bit per pixel	8 bit per pixel			1 bit per pixel

51	The character string “xxxxxxyyzzzz” can be converted and compressed to	6x2y4z	8xy4z	6x2y4z	12xyz			6x2y4z
52	The distance between the center of the dot patterns is called the __________ of the color CRT	Unit	Pitch	Persistence	vector			Pitch
53	In a 3-bit plane frame buffer 011 represents __________ color	Red	Cyan	Black	Green			Cyan
54	The light given off by the phosphor during exposure to the electron beam is known as	Fluorescence	Phosphorescence	Persistence	Durability			Fluorescence
55	The color of the light and time period may ______ from one type of phosphor to another	remain same	Slight deviation	No difference	Vary			Vary

56	The technique for continuous-tone reproduction without sacrificing spatial resolution is called ________ error diffusion	Roy-Steinberg	Joycox-steinberg	Simson-steinberg	Floyd-steinberg			Floyd-steinberg
57	In a 3-bit plane frame buffer the color of magenta is represented by __________	101	100	111	11			101
58	The dots are arranged in a pattern that forms a ____ screen angle with the horizon	35°	180°	45°	90°			45°
59	In practice, news paper halftones use __________ dpi	120 – 200	20 – 40	40 – 60	60 – 80			60 – 80
60	If Blue is represented a 001 then yellow is represented as __________	1	10	101	110			110

KARPAGAM ACADEMY OF HIGHER EDUCATION
DEPARTMENT OF COMPUTER SCIENCE, CA & IT
II B.Sc CS (Batch 2016-2019)
COMPUTER GRAPHICS
PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS
ONLINE EXAMINATIONS
UNIT II

Sno	Questions	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Answer
1	One type of digitizer is the ______	Graphics tablet	joystick	dataglove	keyboard			Graphics tablet
2	Drawings, graphs, color and black-white photos or text can be stored for computer processing with an _______	keyboard	Image Scanners	joystick	mouse			Image Scanners
3	_______ allow displayed objects or screen positions to be selected with the touch of a finger	Trackball	Joystick	Touch panels	data glove			Touch panels
4	A ______ is a ball that can be rotated with the fingers or palm of the hand	Trackball	Joystick	mouse	data glove			Trackball

5	The ________ is an efficient device for inputing such non graphic data as picture labels associated wit a graphic display.	joystick	data glove	keyboard	mouse			keyboard
6	A______ provides six degrees of freedom	Trackball	Joystick	space ball	data glove			space ball
7	The ______ input can be used to initiate graphics operations or to enter data	Voice systems	Joystick	space ball	data glove			Voice systems
8	______ devices are used to select screen positions by detecting the light coming from points on the CRT screen	Pencil shaped	Joystick	Pen shaped	data glove			Pencil shaped
9	______ is an output device.	Trackball	Joystick	space ball	Printer			Printer
10	What is the disadvantage of the light pen?	It's shape	They cannot detect positions	Accurate reading	Cannot detect positions within black areas			Cannot detect positions within black areas
11	_________ is used in graphics workstation as input devices to accept voice commands.	Touch panels	Speech recognizers	Only a	All of these			Speech recognizers

12	What voice the use of voice system?	To initiate graphics operation	To enter data	Neither a nor b	Both a and b			Both a and b
13	When a voice command is given, the system searches the ______________ _for a frequency-pattern match.	Memory	Input data	Dictionar y	Hard disk			Dictionary
14	The device which is designed to minimize the background sound is	Micropho ne	Digitizers	Data glove	Joy stick			Microphone
15	The quality of a picture obtained from a device depends on	Dot size	Number of dots per inch	Number of lines per inch	All the above			All the above
16	Which of the following device is not the input device?	Trackball and space ball	Data glove	Impact printers	Keyboard			Impact printers
17	Which device contains thumbwheel, trackball and a standard mouse ball?	Z mouse	Joystick	Mouse	Trackball			Z mouse
18	Virtual reality, CAD, and animations are the application of	Z mouse	Digitizers	Data tablets	Image scanners			Z mouse
19	The most widely used input device is the _____	Keyboard	Modem	Monitor	Printer			Keyboard

20	The term used to define all input and output devices in a computer system is ________	Monitor	Software	Shared resources	Hardware			Hardware
21	Information that comes from an external source and is fed into computer software is called _____	Input	Output	Throughput	Reports			Input
22	The most common method of entering text and numerical data into a computer system is through the use of a _____	Plotter	Scanner	Printer	Keyboard			Keyboard
23	Which of the following is not an output device?	Plotter	Printer	Scanner	Monitor			Scanner
24	Devices that let the computer communicate with you	Input	Output	Type	Print			Output

25	Devices that allow you to put information into the computer	Input	Output	Type	Print			Input
26	Using output devices one can _____	Input data	Store data	Scan data	View or print data			View or print data
27	The wheel located between the two standard buttons on a mouse is used to _____	Click on web pages	Shutdown	Click and select items	Scroll			Scroll
28	A mouse,touch screen, and a trackball are all examples of _____________	Scanning Device	Voice-Input Devices	Pointing Device	Output Device			Pointing Device
29	The input device that has the monitor screen covered with a flexible layer is the	Touch Screen	Light Pen	Digitizer	Mouse			Touch Screen

30	The device that would be used to read a mark-sense answer sheet for this test would be	Image Scanner	Bar-code reader	Facsimile machine	Character or mark recogniti on device			Character or mark recognition device
31	The most popular input device for computer games is called	Trackball	Touch Screen	Joystick	Pointing stick			Joystick

KARPAGAM ACADEMY OF HIGHER EDUCATION
DEPARTMENT OF COMPUTER SCIENCE, CA & IT
II B.Sc CS (Batch 2016-2019)
COMPUTER GRAPHICS
PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS
ONLINE EXAMINATIONS

Sno	Questions	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Answer
1	In ________________ coordinate system, the points are addressed by x and y coordinates	absolute	relative	Cartesian	basic			Cartesian
2	The number of visible distinct dots displayed in a unit area on the screen is called _________________	resolution of the screen	revolution of the image	screen size	window size			resolution of the screen
3	________ is an incremental computing techniques.	clipping algorithm	Bresenhams line drawing algorithm	display technique	interlacing			Bresenhams line drawing algorithm
4	A line segment starts and finishes at ___________________ points	left	right	middle	addressable			addressable

5	To maintain a constant density in a line, dots are ________________ spaced	equally	less	more	not equally			equally
6	The digital differential analyzer generates lines from __________________ equation	iterative	integration	differential	simple			differential
7	The disadvantage of DDA is ____________	it is an incremental method	error accumulates due to limited precision in floating point representation	it uses floating point addition	it performs calculation at each step			error accumulates due to limited precision in floating point representation
8	Circle exhibits _____ symmetry	bilateral	four way	eight way	linear			eight way
9	In Circle drawing algorithm if D(T)+D(S) is __________ choose S (point inside the true circle)	positive	negative	zero	one			positive
10	_________ is a process of "colouring in" a definite image area.	colour model	scan conversion	region filling	anti aliasing			region filling

11	The mathematical function __________ returns the largest integer that is less than or equal to the argument.	Ceil( )	Round( )	Int( )	Floor( )			Floor( )
12	In the equation of line y=mx+b, _______ is called the slope of the line	x	y	b	m			m
13	The region described as the totality of pixels that comprises it is called _________	boundary defined region	interior defined region	pixel defined region	boundary region filling			interior defined region
14	The collection algorithms used to fill the the interior defined region is called _______	boundary fill algorithm	interior fill algorithm	pixel filling algorithm	flood fill algorithm			flood fill algorithm
15	The starting pixel to fill a region is called ______	index	initial	seed	neighbor			seed

16	__________ algorithm is used to fill regions that are geometrically defined.	boundary fill algorithm	interior fill algorithm	scan line algorithm	flood fill algorithm			scan line algorithm
17	In Bresenham's algorithm floating point ________ is avoided.	addition	subtraction	multiplication	division			division
18	In scanline algorithm the Edges in the edge list becomes ________ when the y co-ordinate of the current scanline matches their Y-min value	active	inactive	minimum	maximum			active
19	The data structure used in scanline algorithm is called ____________	linked list	edge list	vertices	matrix			edge list
20	If a regions "interior” is connected to its "exterior" pixel then it forms a ________ boundary	staircase	connected	continuous	discontinuous			discontinuous

21	In circle generation the theta value is stepped from Theta to ________	Pi	Pi/2	Pi/4	Pi/8			Pi/4
22	Some monitors use a technique called ____________ to double their refreshing rate.	flickering	interlacing	doubling	persistence			interlacing
23	Bresenham's circle generating algorithm will take reflections of _________	Two octets	One octet	Three octets	Four octets			One octet
24	Eight-way symmetry is used by reflecting each calculated point around each ___ axis	35°	180°	45°	90°			45°
25	In Bresenham's circle generating algorithm, if(x,y) is the current pixel position then the x-value of the next pixel position is	X	X – 1	X + 1	X + 2			X + 1

26	In Bresenham's circle generating algorithm, if(x,y) is the current pixel position then the y-value of the next pixel position is __________	Y or y + 1	Y alone	Y + 1 or y - 1	Y or y – 1			Y or y – 1
27	The property that adjacent (neighbouring) pixels are likely to have the same characteristics is called	Area coherence	Spatial coherence	Scan line coherence	Pixel coherence			Spatial coherence
28	The property that adjacent pixels on a scan line are likely to have the same characteristics is called	Area coherence	Spatial coherence	Scan line coherence	Pixel coherence			Scan line coherence
30	The design style of set of character is referred to as its	Typeface	Font size	Font style	None of the above			Typeface
31	Character sizes approximately ranges to	1/12 inch	¾ inch	2/5 inch	½ inch			1/12 inch

32	The technique of using a minimum number of intensity levels to obtain increased visual resolution is called _______	Dithering	Depth cueing	Rendering	Halftoning			Halftoning
33	In Bresenham's line generating algorithm, the decision variable is _	D(T)+D(S)	Δx(s-t)	Both a and b	None of the above			D(T)+D(S)
34	In Bresenham's circle generating algorithm, the decision variable is _	D(T)+D(S)	Δx(s-t)	Both a and b	None of the above			D(T)+D(S)
35	Character sizes approximately ranges to	10 point	11 points	12 points	14 points			12 points
36	A _______ is a small raster containing the relative locations of the pixels that are used to represent the character	Frame buffer	Mask	Display	Shadow			Mask

37	In Line display of characters _________ test is used to determine the intersecting lines	Integration	Black box	Unit	Minmax			Minmax
38	When _________ _____ is used for a picket fence problem, the distance between pickets are kept close to their true distance	Private aliasing	local aliasing	Global aliasing	Public aliasing			local aliasing
39	When _________ _____ is used for a picket fence problem, the overall length of the picket fence is approximately correct	Private aliasing	local aliasing	Global aliasing	Public aliasing			Global aliasing
40	_________ __ occurs when an object is not aligned with, or does not fit into, the pixel grid properly.	Staircase	Picket Fence problem	Unequal Brightness	Asymmetric location			Picket Fence problem

41	A ____________ technique works on the true signal in the continuous space to derive proper values for individual pixels	Post-filtering	Pre-filtering	Area filtering	Super filtering			Pre-filtering
42	A ____________ technique takes discrete samples of the continuous signal and uses the samples to computer pixel values	Post-filtering	Pre-filtering	Area filtering	Super filtering			Post-filtering
43	Pixel phasing is a _______________-based anti-aliasing technique	Mindware	Humanware	Hardware	Software			Hardware
44	The filter with its weight values conforming to a 2D Gaussian distribution is called	Pre Filter	Guassian filter	Weighted Filter	2D Filter			Guassian filter
45	___________ is a post-filtering technique	Super sampling	Area sampling	Low pass sampling	High pass sampling			Super sampling

46	When the character is represented by the on pixel in a bilevel pixel grid It is called________	Bitmap font	Outline font	Italics Font	Courier Font			Bitmap font
48	__________ representation is simple and effective since Characters are defined in already scan converted form.	Outline font	Italics Font	Courier Font	Raster Font			Raster Font
49	In ___________ representation of characters, lines and arcs define a character.	Outline font	Italics Font	Vector Font	Raster Font			Outline font
50	Outline fonts are otherwise called _______	Italics Font	Vector Font	Raster Font	Bitmap font			Vector Font

51	In using methods like Bresenhams algorithm to scan convert arcs there is a danger of missing the end points of the arc. This may result in ___________	infinite loop	ellipse	staircase effect	floating point calculation			infinite loop
52	______ is a arc with two lines drawn from the center to the end points of the arc.	Circle	Sector	Ellipse	Curve			Sector
53	Ellipse exhibits __________ way symmetry.	two	four	six	eight			four
54	Polynomial method of scan converting an ellipse is inefficient because it uses __________	floating addition and subtraction	logarithmic calculations	square and square root operations	integration and differentiation			square and square root operations
55	In Midpoint algorithm, we split the ellipse into two parts at a point Q where __________	major axis = minor axis	x and y coordinate values are equal	x and y coordinate values are distinct	slope of the curve = -1			slope of the curve = -1

56	In midpoint ellipse algorithm, if the decision parameter is greater than zero the midpoint is ________ the curve	inside	outside	equidistant from	above			outside
57	In midpoint ellipse algorithm, if the decision parameter p< 0 the midpoint is ________ the curve	inside	outside	equidistant from	middle			inside
58	For scan converting of ellipse, the first part of the curve the x and y values are obtained by	x is incremented and y is decremented	y is decremented and x is chosen the point close to the true curve	x is incremented and y is chosen the point close to the true curve	both x and y are incremented			x is incremented and y is chosen the point close to the true curve
59	For scan converting of ellipse, the second part of the curve the x and y values are obtained by	x is incremented and y is decremented	y is decremented and x is chosen the point close to the true curve	x is incremented and y is chosen the point close to the true curve	both x and y are incremented			y is decremented and x is chosen the point close to the true curve

60	In trigonometrically defining the ellipse x= a cos θ + h and y = b sin θ + k; then the point (h,k) is the ___________ of the ellipse.	major axis	minor axis	center	point on the curve			center

KARPAGAM ACADEMY OF HIGHER EDUCATION
DEPARTMENT OF COMPUTER SCIENCE, CA & IT
II B.Sc CS (Batch 2016-2019)
COMPUTER GRAPHICS
PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS
ONLINE EXAMINATIONS
UNIT IV

Sno	Questions	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Answer
1	The simulated spacial manipulation of objects is called __________	scan conversion	transformation	distortion	clipping			transformation
2	To position several objects defined in its own coordinate system into a common scene we need ______	flickering	transformation	distortion	clipping			transformation
3	When the object is transformed relative to a stationary coordiante system or back ground it is called ______________ transformation	geometric	coordinate	instance	composite			geometric

4	When the object is held stationary and the coordiante system is transformed it is called _______________ transformation	geometric	coordinate	instance	composite			coordinate
5	An Object in a plane is a set of __________	pictures	images	points	lines			images
6	Every point P in a 2D coordinate system consists of ____________	an x value	a y value	an ordered pair x,y	z axis			a y value
7	Which of these is not a basic type of transformation	translation	rotation	scaling	clipping			clipping
8	In __________ the object is displaced a given distance and direction form its original position	translation	rotation	scaling	clipping			translation
9	v = tx I + ty J is called the __________	translation vector	translation matrix	translation coordinates	translation constants			translation vector

10	Applying the translation, v = tx I + ty J to a point P(x,y) the new point P' has coordinate values____________	x+tx,y-ty	x/tx,y/tx	x+tx,y+ty	x.tx,y.ty			x+tx,y+ty
11	If the direction of rotation is counterclockwise then angle is said to be __________	positive	negative	equal to zero	Not equal to zero			positive
12	If the direction of rotation is _____ then angle is said to be negative	clockwise	anticlockwise	counter clockwise	no rotation			clockwise
13	___________ is a process of altering the dimension of an object	translating	rotation	scaling	Mirror reflection			scaling
14	____________ is a special case of scaling	translating	rotation	scaling	Mirror reflection			Mirror reflection

15	In Scaling transformation S(sx,sy), sx and sy are called ____	scaling vector	scaling matrix	scaling constants	scaling axis			scaling constants
16	Applying Scaling to a point P(x,y) the new point P' has coordinate values____________	x.sx,y.sy	x+sx,y+sy	x.1/sx,y.1/sx	x-xs,y-sy			x.sx,y.sy
17	In Scaling transformation the only point that remains fixed is the __________	origin	center of the image	point to which scaling is applied	center of the pixcel			origin
18	A scaling constant greater than 1 results in _______________ of the object	compression	distortion	magnification	dislocation			magnification
19	A scaling constant less than 1 results in _______________ of the object	compression	distortion	magnification	dislocation			compression

20	If both scaling constants have the same value then the transformation is said to be _______________	perspective	dimetric	homogeneous	orthographic			homogeneous
21	Applying Mirror reflection Mx about x axis to a point P result in P' = Mx(P) where	x=x+Mx,y =y+Mx	x=x,y= -y	x= - x, y = - y	x= -x y= y			x=x,y= -y
22	_______________ is the opposite operation performed by the transformation	viewing transformation	scaling transformation	Instance transformation	Inverse transformation			Inverse transformation
23	Complex transformations can be built by _____	addition of transcormations	composition of functions	Instance transformation	Inverse transformation			composition of functions
24	___________ representaion of transformations is very useful to build complex transformation	scalar	vector	matrix	linked list			matrix

25	Composition of transformations is brought about by ___________	addition of matrices	inverse of matrix	multiplication of matrices	subtraction of matrices			multiplication of matrices
26	The process of multiplying basic transformation matrices is called ___________	addition of matrices	composition of functions	Instance transformation	concatenation of matrices			concatenation of matrices
27	The resultant matrix after concatenation of matrices is called ____________	Resultant matix	Composite Transformation Matrix	Final Transformation Matrix	Unit Matrix			Composite Transformation Matrix
28	CTM stands for	Common transformation matrix	Composite Transformation Matrix	Coordinate transformation Matrix	control transformation			Composite Transformation Matrix
29	_______________ transformation is used to create an instance of an object in the picture coordinat	geometric	coordinat	instance	composite			instance

30	Instance transformation is applied when the picture is composed of	several subpictures	several dimensions	several transformations	several points			several subpictures
31	Transforamtion in a plane is called ________ ________	two dimensional transforamtion	three dimensional transformation	viewing transformation	instance transformation			two dimensional transforamtion
32	________ ________ is the Master coordinate system	World coordinate system	device corodinate system	viewing coordinate system	cartesian coordinate system			World coordinate system
33	Objects are places into a scene by ________ _____ to the World coordinate system.	viewing transformation	modeling transformation	mapping transformation	normalized transformation			modeling transformation
34	The rectangular area with its edges parallel to the WCS is called ________ __	Raster	viewport	window	display device			window

35	The rectangular area with its edges parallel to the NDCS is called ___________	Raster	viewport	window	display device			viewport
36	_______________ is a device independent tool to describe the display area.	World coordinate system	Normalised device corodinate system	viewing coordinate system	cartesian coordinate system			Normalised device corodinate system
37	In NDCS the 1 x 1 unit square at the origin of the coordinate system defines the display area of the ___________	virtual display device	window	viewport	workstation.			virtual display device

38	The process that converts object coordinates in WCS to normalized device coordinates is called ________________	viewing transformation	modeling transformation	mapping transformation	normalization transformation			normalization transformation
39	Normalization transformation is also called ________________	window to viewport mapping	mapping transformation	workstation transformation	clipping			window to viewport mapping
40	The process that maps normalised device coordinates to discrete device/ image coordinates is called _________	viewing transformation	modeling transformation	workstation transformation	normalization transformation			workstation transformation

41	Normalization and Workstation transformations collective known as __________	viewing transformation	modeling transformation	window to viewport mapping	mapping transformation			viewing transformation
42	Workstation transformation is dependent on the ________________ of the display device / frame buffer	height	dimension	color	resolution			resolution
43	When the whole display area of the virtual device is mapped to a physical device that does not have a 1/1 aspect ratio it may introduce _________	flickering	clipping	transformation	geometric distortion			geometric distortion

44	_______________ is used to specify a localized view in the WCS	Window	Apparture	Workstation	Viewport			Window
45	______________ eliminates the objects or its portion that are nor visible through the window	flickering	clipping	transformation	geometric distortion			clipping
46	The window that is used to clip the objects is called ___________	Workstation	Viewport	Clipping Window	Apparture			Clipping Window
47	_____________ is specified by four World coordinates.	Workstation	Viewport	Window	Raster			Window
48	_____________ is specified by four Normalised device coordinates	Workstation	Viewport	Window	Raster			Viewport

49	If Xmin, Xmax,Ymin and Ymax define the clipping window a point (x,y) is said to be __________the clipping window if Xmin<x<Xmax but not Ymin<y<Ymax	inside	outside	on the diagonal	on the left border of			outside
50	If Xmin, Xmax,Ymin and Ymax define the clipping window a point (x,y) is said to be __________ of the clipping window if x=Xmin and y=Ymin	at the bottom left corner	on the top border	on the diagonal	on the right border			at the bottom left corner

51	Lines that ____________ the clipping window are either completely inside or completely outside the window	intersect	parallels the borders of	doesnot intersect	above or below			doesnot intersect
52	If both the endpoints of the line lies inside the clipping window the line is _________	a clipping candidate	uneven	visible	invisible			visible
53	The lines intersecting the edges of the clipping window are called _____	clipping candidate	segments	visible	invisible			clipping candidate

54	In Midpoint Subdivision method of line clipping if maximum number of pixels on line is (M =) 1024, we need atmost (N =) ___________subdivisions to clip it.	5	10	20	15			10
55	In Cohen Sutherland Algorithm, if the bitwise logical AND of the region codes of the endpoints is 0000, then the line is _________	a clipping candidate	visible	invisible	a clipping coordinate			a clipping candidate

56	A polygon is called _________ if the line joining any two interior points of the polygon lies completely insdie the polygon.	plannar	polyline	concave	convex			convex
57	A polygon is said to be ______ if a tour of vertices in the given order produces a clockwise circuit.	concave	convex	positively oriented	negatively oriented			negatively oriented
58	The polygon to be clipped is called _____________	clipping polygon	subject polygon	plannar polygon	convex polygon			subject polygon

59	At the beginning of the Graphics Pipeline we have _________ stored in application specific data structures	colour attributes	viewing parameters	object data	scan conversion techniques			object data
60	In animated sequence of images ___________ is used to reduce flickering	panning	Interlacing	high resolution monitor	Double bufering			Double bufering
61	In Weiler-Atherton Algorithm, If the edge __________, record the intersection point and continue to trace the subject polygon.	enters the clipping window	leaves the clipping window	intersects another edge	intersects the subject polygon			enters the clipping window

62	If a point P is to the left of every edge of a positively oriented polygon it is ________ the polygon	on the edge	below	outside	inside			inside
63	In Ploygon clipping the edges that are drawn twice in opposite direction are called ___	extra egdes	parallel edges	positive edges	clipping edges			extra egdes
64	In the 4 bit region code of Cohen Sutherland's line clipping algorithm, the first bi correspon ds to he region ________ the window	left of	right of	above	below			above

65	In Liang-Barsky's Line clipping algorithm if ____________ then the line is parallel to the corresponding boundary.	pk<0	pk>0	pk=0	pk!=0			pk=0

KARPAGAM ACADEMY OF HIGHER EDUCATION

DEPARTMENT OF COMPUTER SCIENCE, CA & IT

II B.Sc CS (Batch 2016-2019)

COMPUTER GRAPHICS

PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS

ONLINE EXAMINATIONS

Sno	Questions	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Answer
1	__________ is defined as a mapping of a point P to its image P' in the view plane.	Transformation	Viewing	Clipping	Projection			Projection
2	The mapping is determined by a line that passes through point P and intersects the view plane, this line is called ______	projector	normal	vetor	axes			projector
3	Which of these are basic types of projections	parallel projection	perspective projection	Both	no projection			Both
4	_________ projection is called converging projectors.	Parallel	Perspective	Isometric	Axonometric			Perspective
5	___________ drawings are characterised by vanishing points.	Parallel	Perspective	Isometric	Axonometric			Perspective

6	The perspective projection gives the illusion that certain set of parallel lines appear to meet at a point called ______	view point	projection point	vanishing point	reference point			vanishing point
7	__________ projection preserves the shape and size of the projected object.	Parallel	Perspective	both	projection point			Parallel
8	__________ algorithm does not suit for 3 dimensional clipping.	Cohen Sutherland algorithm	Bresenham algorithm	Sutherland Hodgman algorithm	DDA			Bresenham algorithm
9	__________ defines the spatial extent that is visible through the rectangular window of the view plane.	view point	view volume	viewport	view region			view volume
10	The view plane is also called ___________	projection plan	view direction	mapping funtion	reference vector			projection plan
11	The Center of projection is called the _____	view point	view volume	viewport	reference vector			view point

12	Three dimensional viewing of objects requires the specification of ________	view plane	reference vector	view volume	view region			view plane
13	The unit normal vector N = n1I + n2J + n3 K , |N| = __________	0	1	2	3			1
14	N = n1I + n2J + n3 K in 3D viewing is called the _____	Reference vector	Translation Vector	Unit Normal vector	Up vector			Unit Normal vector
15	Reference Vector U is also known as _________	Co-ordinate Vector	Translation Vector	Unit Normal vector	Up vector			Up vector
16	The triad formed by vectors Ip, Jp and N is called _____	Cartesian co-ordinate system	Vector Coordinate system	Viewing coordinate system	x & y cordinate			Viewing coordinate system
17	______ coordinate system is choosen because the p and q coordinate axes are superimposed on the display device	Left handed	Right handed	Cartesian	3 dimensional			Left handed

18	In Left handed coordinate system the normal vector N point ________,facing the display.	towards the viewer	away from the viewer	the right side of the viewer	left side of the viewer			away from the viewer
19	In Left handed coordinate system increasing distance away from the observer is measured along _________	Ip	Jp	N	Up			N
20	If the xy plan is the view plane then _________ measures the depth or distance of the point from the view plan.	x coordinate	y coordinate	z coordinate	orgin			z coordinate
21	If the view plan is the xy plane then Ip = I, Jp = J and N = __________	Np	-K	-N	z			-K

22	The right handed world coordinates (x,y,z) can be changed to left handed view plane coordinates (x',y',z') by performing the transformation where x'=x,y'=y and z' = _________	z	-z	k	-k			-z
23	The _________ bounds a region in world coordinate space that is clipped and projected to the view plan.	view point	view volume	viewport	view region			view volume
24	The region bounded by the view volume is__________ to the viewplane.	transformed	mapped	clipped	clipped and projected			clipped and projected
25	For a ___________, the view volume corresponding to the given window is a semi-infinite pyramid.	parallel view	prespective view	Axanometric view	Orthographic view			prespective view

26	For parallel projected views, the view volume is _____	a semi infinite pyramid	finite parallelog ram	infinite parallelopi ped	finite polygon			infinite parallelopiped
27	The view volumes are _______ in extent	finite	infinite	invisible	fixed			infinite
28	For perspective views, very distant objects appear as _________	a semi infinite pyramid	infinite parallelopi ped	disjointed structure	indistingui shable spots			indistinguishable spots
29	For perspective views, objects very close to the center of projection appear as _________	disjointed structure	indistingui shable spots	infinite parallelopi ped	finite polygon			disjointed structure
30	A finite volume is delimited by front and back clipping planes _________ to the view plane	perpendic ular	parallel	similar	normal			parallel
31	In __________ clipping strategy, clipping is done against the original view volume	canonical	normalize d	tranforma l	direct			direct

32	In __________ clipping strategy, a normalizing transformation is applied to the original view volume before clipping is done.	canonical	normalized	tranformal	direct			canonical
33	When normalizing transformation is applied to the original view volume it is called __________ view volume	canonical	normalized	tranformal	direct			canonical
34	The canonical view volume for a parallel projection is the _________	infinte pyramid	infinte parallelopiped	unit cube	truncated pyramid			unit cube
35	The canonical view volume for a perspectivel projection is the _________	infinte pyramid	infinte parallelopiped	unit cube	truncated pyramid			truncated pyramid

36	After clipping in viewing coordinates, the resulting structure is projected onto the ________	clipping window	projector	World Coordinate system	screen projection plane			screen projection plane
37	In 3D graphics pipeline, viewing transformation and projection are carried out according to the _____ parameters set by the application.	viewing	projecting	transforming	colour			viewing
38	In 3D graphics pipeline, the result of projection is mapped to workstation viewport via ________	3D viewing transformation	2D viewing transformation	scaling transformation	Rotation transformation			2D viewing transformation
39	Which of these is a complex geometric form.	point	line	curves	polygon			curves
40	_________ and _________ are the basic building blocks of computer graphics.	points and lines	lines and curves	points and curves	polylines and curves surface patches			points and lines
41	________ is a chain of connected line	points	curves	quadric	polylines			polylines

42	__________ is a closed polyline	points	curves	quadric surfaces	polygon			polygon
43	A _________ is a polygon in which all vertices lie on the same plane	convex polygon	concave polygon	planar polygon	curves			planar polygon
44	The _________is also called polygonal net or polygonal mesh.	planar polygon	wireframe model	polyline	quadric surface			wireframe model
45	A _________ is a closed polygonal net in which each polygon is planar.	curved surface	polyline	polyhedron	quadric surface			polyhedron
46	__________ is a method of representing a polygonal net model.	Explicit vertex and edge listing	polyline	quadric surface	polyhedron			Explicit vertex and edge listing
47	The polygons are called the __________ of the polyhedron	edges	vertices	faces	sides			faces
48	Which of these is complex in wire frame models	constructing wireframe models	representing curved surfaces	applying geometic transformation	applying coordinate transformation			representing curved surfaces

49	A model constructed using solid objects as building blocks is called ______	Wirefram e modeling	Solid object modeling	Block modeling	object modeling			Solid object modeling
50	________ is a process of building model by assembling simpler objects.	Additive modeling	Subtractiv e Modeling	Destructiv e modeling	Constructi ve modeling			Additive modeling
51	________ is a process of removing pieces from a given object to create a new object.	Additive modeling	Subtractiv e Modeling	Destructiv e modeling	Constructi ve modeling			Subtractive Modeling
52	A curve is specified by _________	polylines	curved surface pathes	a set of control points	two endpoints			a set of control points
53	The Control points control the _________ of the curve.	length	shape	slope	width			shape
54	Curves are generally _________ with respect to any coordinate system.	single valued	bi-valued	multi- valued	axis level			multi-valued

55	If the movement of the control point affects the shape of the curve only in a small neighbourhood of the control point it is called ________________	Coordinate independence	Variation diminishing effect	Versatility	Local controllability.			Local controllability.
56	The perception of color arises from ___________ entering our visual system.	image	view volume	light	color			light
57	Light is a _________ energy.	electrical	mechanical	kinetic	electromagnetic			electromagnetic
58	The wavelength of the visual light ranges form ________nanometer.	100 to 500	400 to 700	30 to 90	600 to 1400			400 to 700
59	Which of these is the characteristics of light	brightness ,hue,saturation	purity	saturation	luminance			brightness ,hue,saturation
60	__________ coresponds to the physical property called luminance.	brightness	hue	saturation	purity			brightness

61	____________ property distinguishes a white light from a red or green light.	brightness	hue	saturation	purity			hue
62	__________ coresponds to the physical property called excitation purity.	brightness	hue	saturation	luminance			saturation
63	__________ describes the degree of vividness.	brightness	hue	saturation	luminance			saturation
64	Hue corresponds to another physical property called the _________	excitation purity	dominant wavelength	brightness	luminance			dominant wavelength
65	The receptor cells in the retina of the eye, that are sensitive to color is __________	rods	cones	cubes	lines			cones
66	The receptor cells in the retina of the eye, that are sensitive to color is __________	rods	cones	cubes	lines			rods

67	The international Commission on Illumination defined the __________ colour model.	RGB	CMY	XYZ	YIQ			XYZ
68	XYZ colors were the result of an ___________ ______ transformation applied to three real primaries	Wavelet	Gamut	3 dimentional	Affine			Affine
69	___________ of human eye coresponds to the eye's response to light of constant luminance.	dominant wavelength	luminous efficiency	excitation purity	Wavelet			luminous efficiency
70	The curved triangular figure of CIE Chromaticity Diagram encompases __________	Whole Light spectrum	only the white light	all perceivable colors and their luminance	all perceivable colors by ignoring the luminance			all perceivable colors by ignoring the luminance
71	___________ _____ refers to NTSC	National Telecommunication System committee	National Telecasting system committee	National Television System committee	National Tel-system Committee			National Television System committee

72	_______________Model , main focus is on the direct impact of the light coming from the light source	Global Illumination	Local Illumination	Specular Illumination	diffuse Illumination			Local Illumination
73	_______________ Model, attempts to include secondary effects as light going through transparent / translucent material and light bouncing from one object surface to another	Global Illumination	Specular Illumination	Local Illumination	diffuse Illumination			Global Illumination
74	In _______________ reflection, light energy from the light source gets reflectd / bounced off equally in all the direction	Global reflection	Local reflection	Specular reflection	diffuse reflection			diffuse reflection

75	_________________ reflection, attempts to capture the characteristics of a shiny or mirror-like surface	diffuse reflection	Local reflection	Specular reflection	Global reflection			Specular reflection
76	In ____________ shading, Instead of color values, normal vector is found Interpolatively	Phong	constant	Gouraud	sutherland			Phong
77	Regular or Irregular surface feature details are colectively referred to as __________	watermark texture	glomming texture	surface texture	scaleable texture			surface texture
78	When an AND operation is used, a texture area with __________ shades will appear unaltered if the original color is white	black	white	red	magenta			magenta

79	When an AND operation is used, a texture area with __________ shades will appear unaltered if the original color is yellow	black	white	red	magenta			red
80	__________ texture is an effective tool when target surface are relatively flat and facing the reference plane	projected	solid	mapping	interpolative			projected
81	In ________________ texture, we can wrap around the surface of an object, stretch or shrink it so as to follow the shape of the object	projected	solid	texture mapping	interpolative			texture mapping
82	___________ is a 3D representation of the internal structure of some nonhomogeneous material.	projected	solid	texture mapping	interpolative			solid

83	A Global illumination model that accounts for the transport of light energy beyond the direct contribution from the light sources	Light Tracing	Ray Tracing	basic Tracing	wave tracing			Ray Tracing
84	Which among the following is NOT, the three components of the surface shading used in several secondary ray	local	reflected	transmitted	refracted			refracted
85	A Vector is defined by its	direction and starting point	direction and end point	only direction	direction and magnitude			direction and magnitude
86	A Ray is determined by its	direction and starting point	direction and end point	only direction	direction and magnitude			direction and starting point