COURSE MATERIAL

IV Year B. Tech I- Semester

MECHANICAL ENGINEERING

AY: 2024-25

CAD/CAM

R20A0328

Prepared by:

Mrs.Indraja Bhadri

Assistant Professor

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF MECHANICAL ENGINEERING

(Autonomous Institution-UGC, Govt. of India)
Secunderabad-500100, Telangana State, India.
www.mrcet.ac.in

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

DEPARTMENT OF MECHANICAL ENGINEERING

CONTENTS

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

VISION

- To establish a pedestal for the integral innovation, team spirit, originality and competence in the students, expose them to face the global challenges and become technology leaders of Indian vision of modern society.

MISSION

- To become a model institution in the fields of Engineering, Technology and Management.
- To impart holistic education to the students to render them as industry ready engineers.
- To ensure synchronization of MRCET ideologies with challenging demands of International Pioneering Organizations.

QUALITY POLICY

- To implement best practices in Teaching and Learning process for both UG and PG courses meticulously.
- To provide state of art infrastructure and expertise to impart quality education.
- To groom the students to become intellectually creative and professionally competitive.
- To channelize the activities and tune them in heights of commitment and sincerity, the requisites to claim the never - ending ladder of **SUCCESS** year after year.

For more information: www.mrcet.ac.in

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

www.mrcet.ac.in

Department of Mechanical Engineering

VISION

To become an innovative knowledge center in mechanical engineering through state-of-the-art teaching-learning and research practices, promoting creative thinking professionals.

MISSION

The Department of Mechanical Engineering is dedicated for transforming the students into highly competent Mechanical engineers to meet the needs of the industry, in a changing and challenging technical environment, by strongly focusing in the fundamentals of engineering sciences for achieving excellent results in their professional pursuits.

Quality Policy

- ✓ To pursuit global Standards of excellence in all our endeavors namely teaching, research and continuing education and to remain accountable in our core and support functions, through processes of self-evaluation and continuous improvement.
- ✓ To create a midst of excellence for imparting state of art education, industry-oriented training research in the field of technical education.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

www.mrcet.ac.in

Department of Mechanical Engineering

PROGRAM OUTCOMES

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1 Ability to analyze, design and develop Mechanical systems to solve the Engineering problems by integrating thermal, design and manufacturing Domains.

PSO2 Ability to succeed in competitive examinations or to pursue higher studies or research.

PSO3 Ability to apply the learned Mechanical Engineering knowledge for the Development of society and self.

Program Educational Objectives (PEOs)

The Program Educational Objectives of the program offered by the department are broadly listed below:

PEO1: PREPARATION

To provide sound foundation in mathematical, scientific and engineering fundamentals necessary to analyze, formulate and solve engineering problems.

PEO2: CORE COMPETANCE

To provide thorough knowledge in Mechanical Engineering subjects including theoretical knowledge and practical training for preparing physical models pertaining to Thermodynamics, Hydraulics, Heat and Mass Transfer, Dynamics of Machinery, Jet Propulsion, Automobile Engineering, Element Analysis, Production Technology, Mechatronics etc.

PEO3: INVENTION, INNOVATION AND CREATIVITY

To make the students to design, experiment, analyze, interpret in the core field with the help of other inter disciplinary concepts wherever applicable.

PEO4: CAREER DEVELOPMENT

To inculcate the habit of lifelong learning for career development through successful completion of advanced degrees, professional development courses, industrial training etc.

PEO5: PROFESSIONALISM

To impart technical knowledge, ethical values for professional development of the student to solve complex problems and to work in multi-disciplinary ambience, whose solutions lead to significant societal benefits.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

www.mrcet.ac.in

Department of Mechanical Engineering

Blooms Taxonomy

Bloom's Taxonomy is a classification of the different objectives and skills that educators set for their students (learning objectives). The terminology has been updated to include the following six levels of learning. These 6 levels can be used to structure the learning objectives, lessons, and assessments of a course.

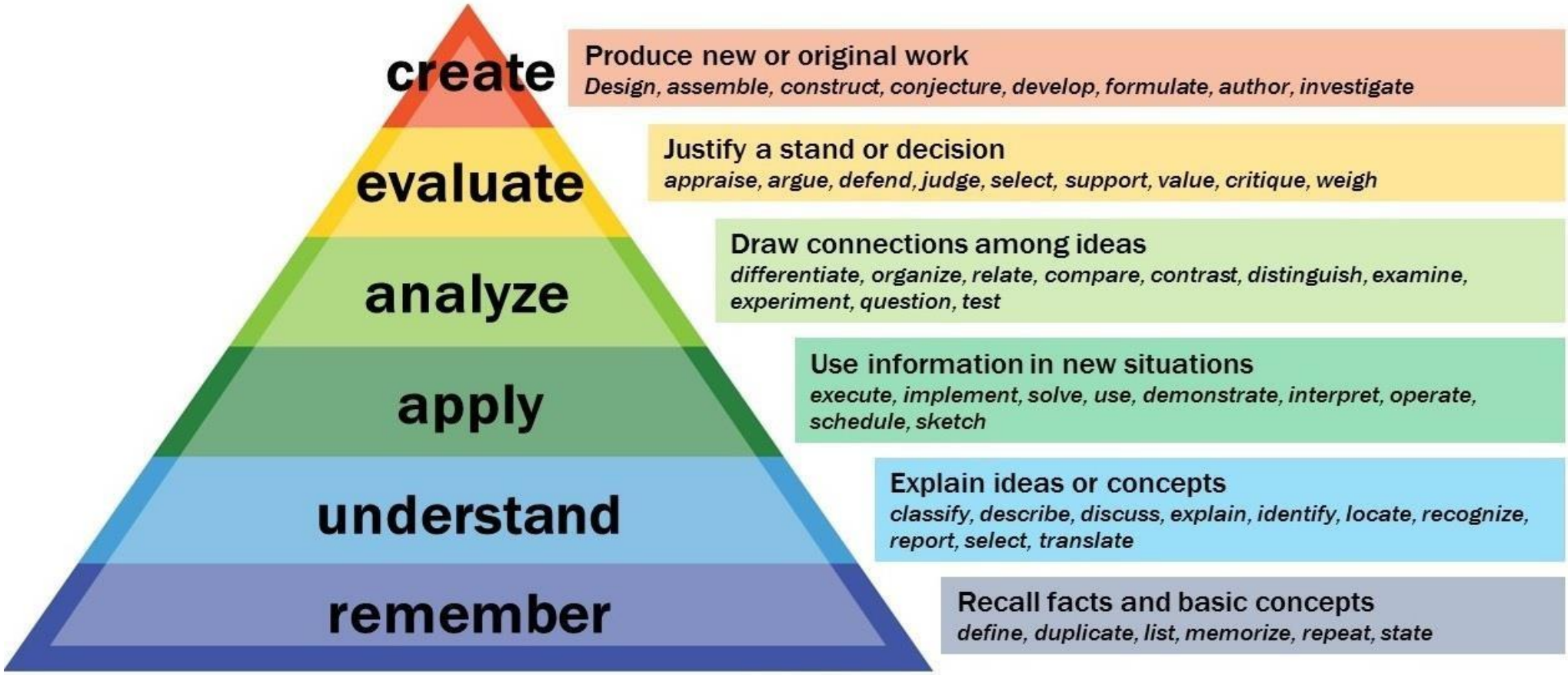
1. **Remembering**: Retrieving, recognizing, and recalling relevant knowledge from long- term memory.
2. **Understanding**: Constructing meaning from oral, written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining.
3. **Applying**: Carrying out or using a procedure for executing or implementing.
4. **Analyzing**: Breaking material into constituent parts, determining how the parts relate to one another and to an overall structure or purpose through differentiating, organizing, and attributing.
5. **Evaluating**: Making judgments based on criteria and standard through checking and critiquing.
6. **Creating**: Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

www.mrcet.ac.in

Department of Mechanical Engineering



MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

IV Year B.Tech. ME- I Sem **L/T/P/C**

3/-/-/3

(R20A0328) CAD/CAM

Course Objectives:

1. To provide an overview of how computers are being used in design, development of man plans and manufacture
2. Understand the Mathematical representations of curves and surfaces used in construction.
3. Understand the Mathematical representations of solids used in geometric construction.
4. Understand the transformation of 2D and 3D parts
5. Understand the algorithm for visualization of various 2D and 3D parts

UNIT–I

Introduction: Computers in Industrial Manufacturing, Product cycle, CAD / CAM Hardware, Basic structure.

Computer Graphics: Display Devices: Cathode Ray Tube, DVST, Raster display, pixel value and lookup table, estimation of graphical memory, LCD, LED fundamentals. Concept of Coordinate Systems: Working Coordinate System, Model Coordinate System, Screen Coordinate System. Graphics exchange standards and Database management systems.

UNIT–II

Curves and Surfaces: Introduction to curve representation, Classification of curves, Line and Curve generation algorithm: DDA algorithm. Synthetic Curves: Concept of continuity, Cubic Spline: equation, properties and blending. Bezier Curve: equations, properties; Properties and advantages of B-Splines and NURBS. Various types of surfaces along with their typical applications.

UNIT–III

Mathematical representation of solids: Geometry and Topology, Comparison of wireframe, surface and solid models, Properties of solid model, properties of representation schemes, Concept of Half-spaces, Boolean operations. Schemes: B-rep, CSG, Sweep representation, ASM, Primitive instancing, Cell Decomposition and Octree encoding

UNIT–IV

Geometric Transformations: Homogeneous representation; Translation, Scaling, Reflection, Rotation, Shearing in 2D; Orthographic and perspective projections.

UNIT–V

VISUAL REALIS M Hidden – Line-Surface-Solid removal algorithms – shading – colouring – Computer animation

TEXT BOOKS:

1. CAD / CAM Theory and Practice / Ibrahim Zeid / TMH Publishers
2. CAD / CAM /A Zimmers&P.Groover/PE/PHI Publishers

REFERENCE BOOKS:

1. CAD / CAM / CIM / Radhakrishnan and Subramanian / New Age Publishers
2. Principles of Computer Aided Design and Manufacturing / Farid Amirouche / Pearson Edu
3. CAD/CAM: Concepts and Applications/Alavala/ PHI Publishers Computer Numerical Control Concepts and programming / Warren S Seames / Thomson Publishers
4. CAD / CAM – P N RAO McGraw Hill Publications

Course Outcomes:

1. Understand the applications of computer in the design and manufacturing.
2. Understand and develop the Mathematical representations of curves used in geometric construction.
3. Understand and develop the Mathematical representations of solids used in geometric construction.
4. Able to get the transformed in 2D and 3D using transformation equations
5. Understand and develop the algorithm for visualization of various 2D and 3D parts

UNIT 1

Fig 1.1: The influence of computers used in manufacturing environment

PRODUCT CYCLE

For any product to be developed, the major areas to be looked into would be designing, manufacturing and marketing. The synchronization between these processes plays a great role in defining product success.

Here CAD/CAM revolutionize the process of the traditional design and cut costs of any mechanical part. CAD/CAM drastically reduces the design time for a new product. This gives it so much edge on the traditional design process that nowadays, traditional design with pen and paper is almost extinct. Automobile industry isthe best example of this improved design time and product cycle.

Product life cycle (PLC) Like human beings, products also have a life-cycle. From birth to death, human beings pass through various stages e.g. birth, growth, maturity, decline and death. A similar life-cycle is seen in the case of products. The product life cycle goes through multiple phases, involves many professional disciplines, and requires many skills, tools and processes. Product life cycle (PLC) has to do with the life of a product in themarket with respect to business/commercial costs and sales measures. To say that a product has a life cycle is to assert three things:

- Products have a limited life,
- Product sales pass through distinct stages, each posing different challenges, opportunities, andproblems to the seller,

Products require different marketing, financing, manufacturing, purchasing, and human resourcestrategies in each life cycle stage.

The Traditional Design Process:

Traditionally, the design process contains the following steps:

1. Recognition of Need
2. Definition of Problem
3. Synthesis
4. Analysis and Optimization
5. Evaluation
6. Presentation

Here is the iteration between step 3 and 4 we analyze, find the fault, then again synthesize, again analyze anditerate and iterate again until we would be able to get satisfactory result. After we analyze and optimize the design again if we are not satisfied after evaluation, we again iterate i.e. we again goes to the synthesis.

After all these steps, we finally make a prototype of the part which we are designing. If that fails or does notfulfill our needs, the whole process will start again.

This is what traditional design process does. Here CAD/CAM comes into the picture. If we use computer for theproblem of synthesis like we are designing the things with the help of a computer, all the analysis and synthesiswill be done on computer and the process consumes much lesser time than a traditional design process.

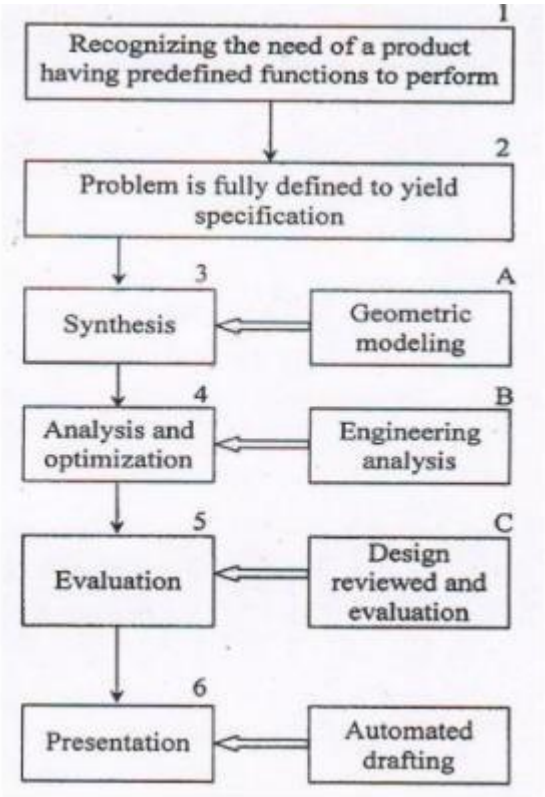


Fig 1.2: Design Process

Stage1: Recognizing the fact that there is a need for a new product for intended function. It may also includethe modification in the existing product.

Stage2: Problem is fully defined in terms of functionality and meeting other requirements such as ergonomic,performance-data, statutory, etc.

Stage3: The design undergoes synthesis, joining its various elements.

Stage4: Product analysis reveals the weaknesses and thus weaknesses can be considered for improvement.This process is repeated until an acceptable Design achieved.

Stage5: The optimized Design is reviewed from the point of view of expected performance. It can be done through proto-type modeling and testing against the set standard.

Stage6: Stages 4 and 5 are repeated until acceptable, optimized design is achieved. These stages are basically iterative in-nature. Iteration depends on the creativeness, ingenuity (skill for devising) and experience of designers and the software (tools) available.

The process (stage 1 and 2) are human dependent while the stages 3, 4, 5 and 6 (four stages) are computerbased (CAD)

(A) Geometric modeling: - It implies the existence of a computer graphics screen and some interaction with thecomputer to generate the geometry and topology of the part.

(B) Engineering analysis: - Communicates with the data base to retrieve the part description and with the userto obtain the design constraints, boundary conditions, and other details of the analysis.

(C) Evaluation module: - Allows the user to check the correctness, manufacturability, and processing details ofthe part.

(D) The drafting and documentation module: - Contains some of both the oldest and newesttechnologies/Computer plotting of engineering drawings.

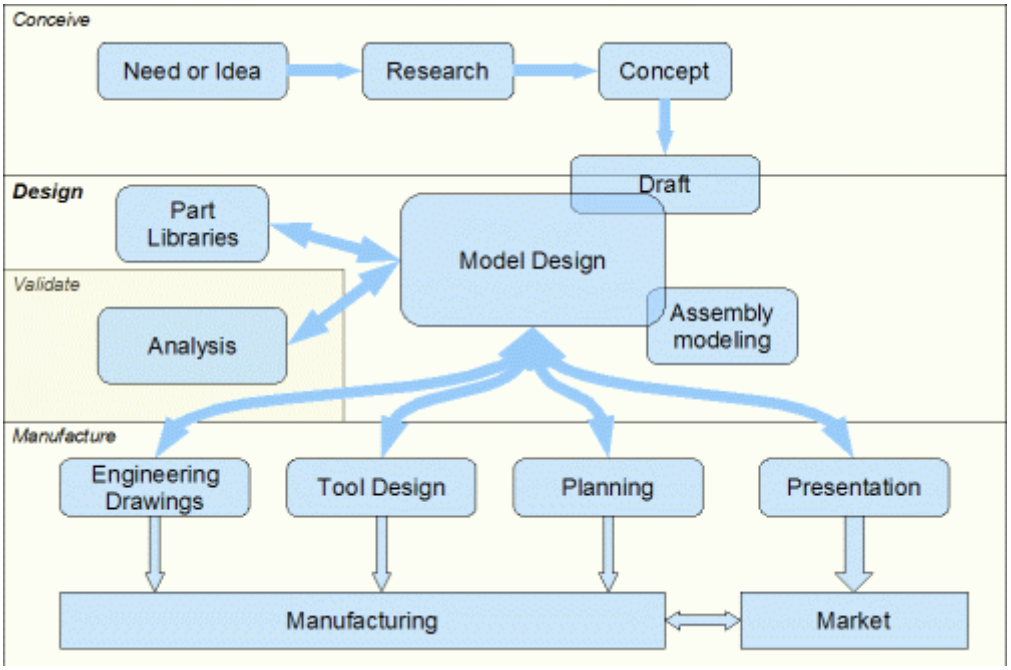


Fig 1.3: Design Process

The scope of CAD/CAM in the operations of a manufacturing firm can be analyzed by examining the variousactivities and functions that must be accomplished in the design and manufacture of a product.

These activities and functions can be referred as the product cycle. A diagram showing the various steps in theproduct cycle is presented in Figure. The cycle is driven by customers and markets which demand the product. It is realistic to think of these as a large collection of diverse industrial and consumer markets rather than one monolithic market. Depending on the particular customer group, there will be differences in the way the product cycle is activated in some cases, the design functions are performed by the customer and the product is manufactured by a different firm. In other cases, design and manufacturing is accomplished by the same firm. Whatever the case, the product cycle begins with concept, an idea for a product. This concept is cultivated, refined, analyzed, improved, and translated into a plan for the product through the design engineering process.

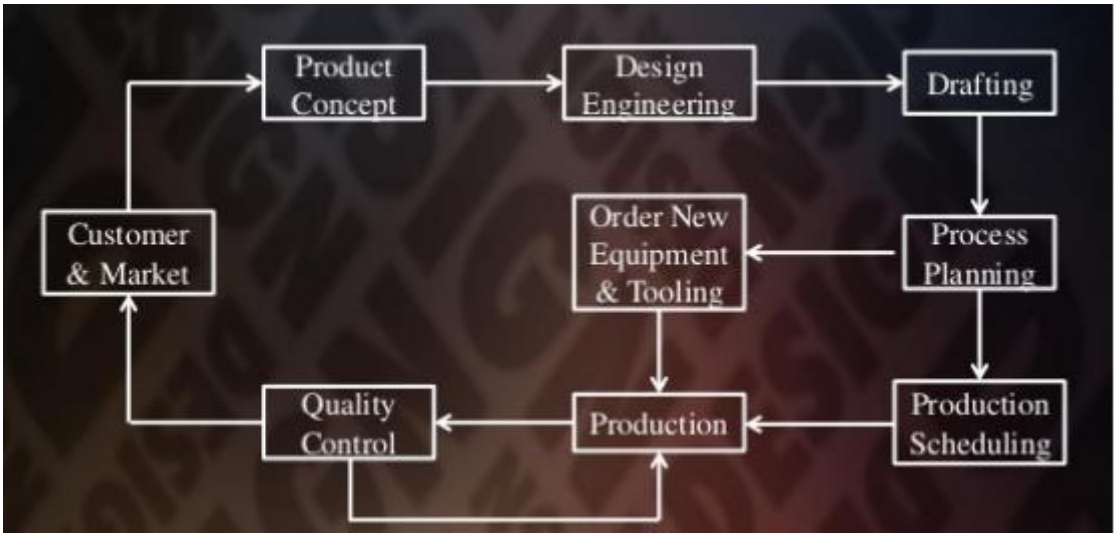


Fig 1.4: Conventional Product Cycle (Design and Manufacturing)

In the design and production operations of a modern manufacturing firm, the computer has become a pervasive, useful, and indispensable tool.

It is strategically important and competitively imperative that manufacturing firms and the people who areemployed by them understand CAD/CAM

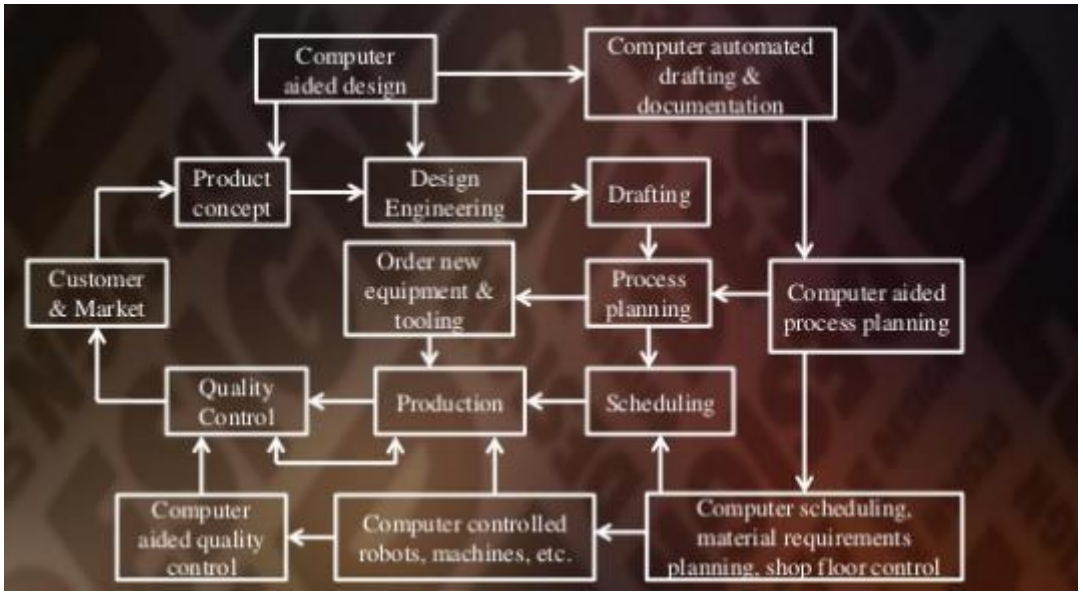


Fig 1.5: Product Cycle revised with CAD/CAM overlaid

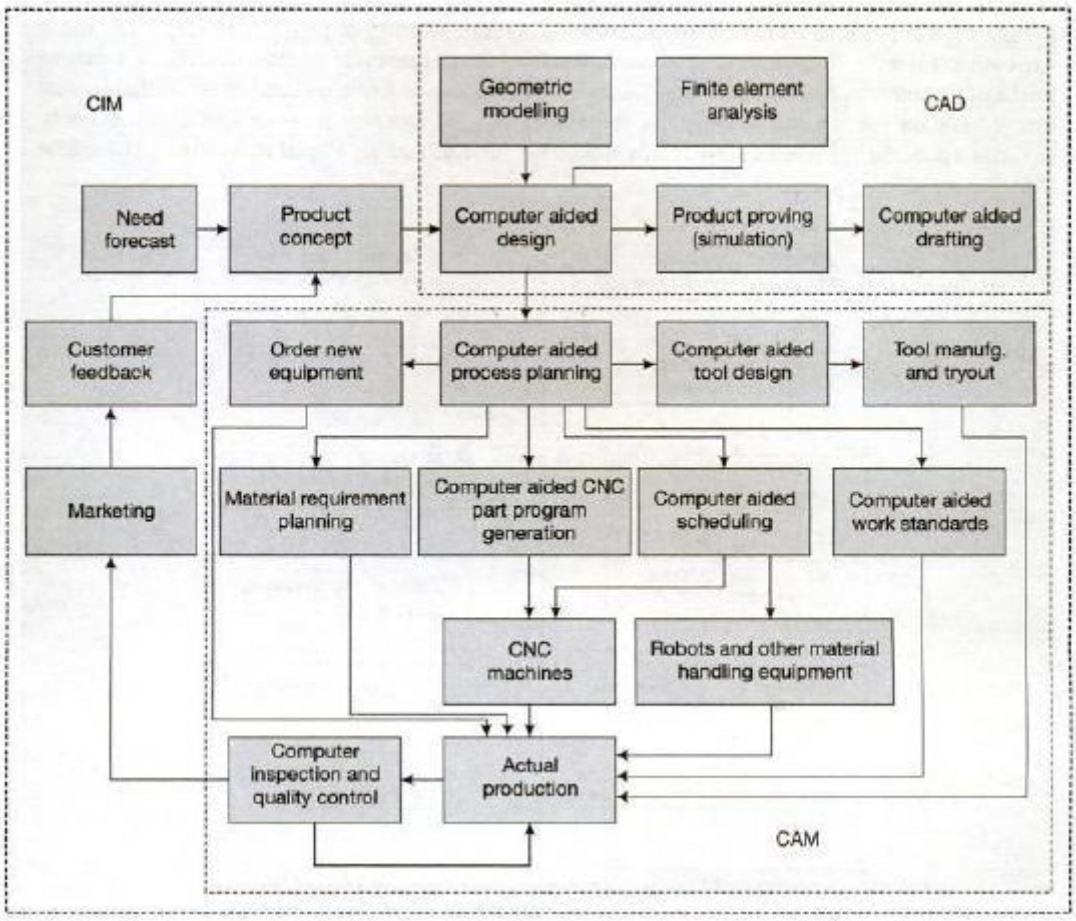


Fig 1.5: The Product Cycle in a computerized manufacturing environment

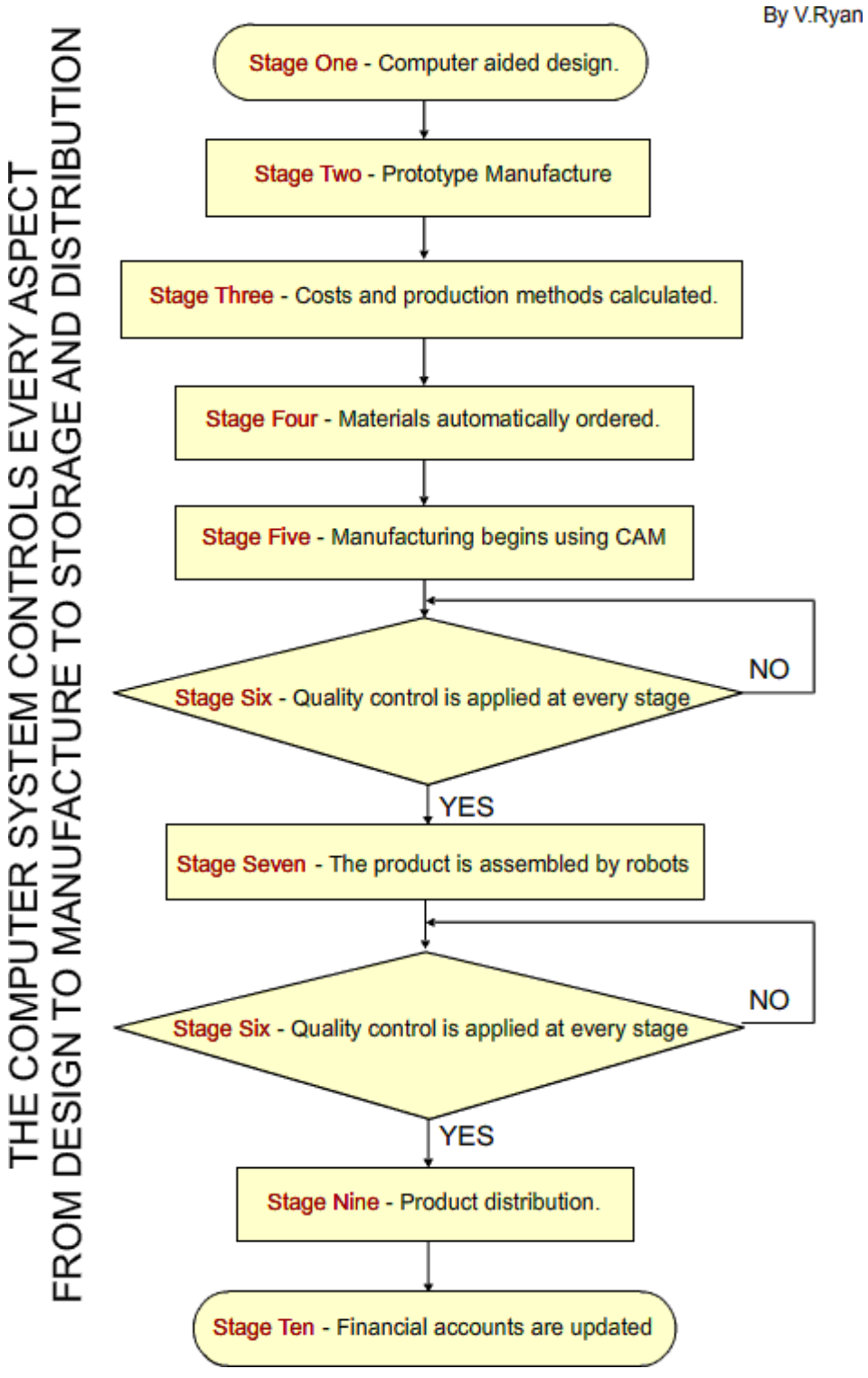


Fig 1.6: 10 key functions of Computer Integrated Manufacturing

CAD/CAM HARDWARE

The developments that have fuelled the growth of the CAD/CAM industry are mostly based on the major developments in microelectronics, which have helped in leapfrogging the capabilities.

The major component of any CAD/CAM system is the basic hardware that goes with the computational aspect,that is, the computer and the associated peripherals that go with it.

Basic Structure

The computing system in operation can be compared to a human being in to of its operating characteristics.The basic configuration of a typical computer system is shown in Fig. 1.7.

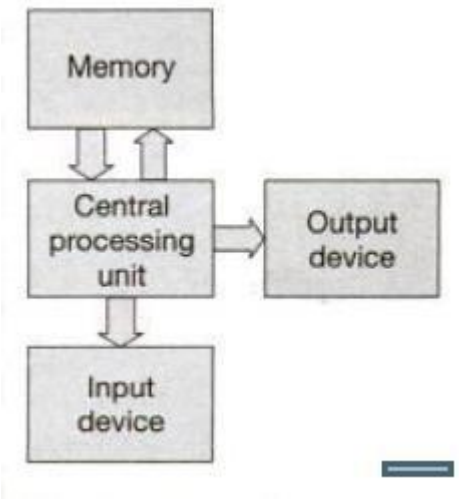


Fig 1.7: Configuration of a typical CAD system

The heart of any computing system is the Central Processing Unit (CPU). It is in the CPU that all the necessary functions of a computer are carried out. The main functions performed in the CPU are arithmetic and logic operations The CPU communicates with the external world through input/output devices or, in short, I/O devices. These are similar to the sensory organs by which a human being maintains contact with the outside environment. These are also collectively called peripheral units. Through an input device, the user would be able to communicate with the CPU, either to give certain data or to control the operation of the CPU. The output device is a means through which the CPU gives the results of the computations.

Another important unit of a computer system is the memory unit. These are the areas where the necessary data or program (sequence of instructions) is stored. The type of memory and its amount determines the capabilities of a typical computing computing system. The present-day computers work on a principle called the stored-program concept. It means that the sequence of operations to be carried out by the CPU is stored in the memory of the computer. The CPU therefore reads an instruction and executes it and continues doing so until it reaches the end of the program. This stored program is called software in computer terminology. Since it is the software, which runs any computer, the computer is as good as the program that is running at anygiven time.

CAD/CAM HARDWARE

The developments that have fuelled the growth of the CAD/CAM industry are mostly based on the major developments in microelectronics, which have helped in leapfrogging the capabilities.

The major component of any CAD/CAM system is the basic hardware that goes with the computational aspect,that is, the computer and the associated peripherals that go with it.

Basic Structure

The computing system in operation can be compared to a human being in to of its operating characteristics.The basic configuration of a typical computer system is shown in Fig. 1.7.

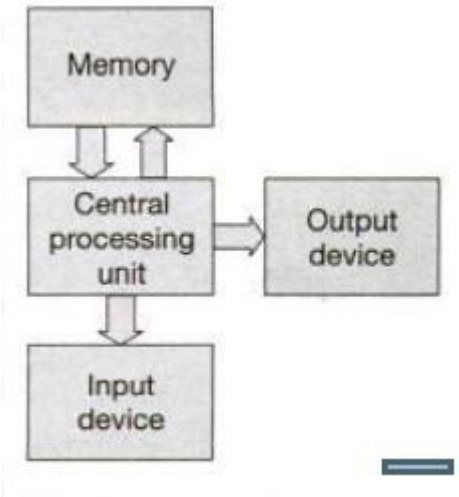


Fig 1.7: Configuration of a typical CAD system

The heart of any computing system is the Central Processing Unit (CPU). It is in the CPU that all the necessary functions of a computer are carried out. The main functions performed in the CPU are arithmetic and logic operations The CPU communicates with the external world through input/output devices or, in short, I/O devices. These are similar to the sensory organs by which a human being maintains contact with the outside environment. These are also collectively called peripheral units. Through an input device, the user would be able to communicate with the CPU, either to give certain data or to control the operation of the CPU. The output device is a means through which the CPU gives the results of the computations.

Another important unit of a computer system is the memory unit. These are the areas where the necessary data or program (sequence of instructions) is stored. The type of memory and its amount determines the capabilities of a typical computing computing system. The present-day computers work on a principle called the stored-program concept. It means that the sequence of operations to be carried out by the CPU is stored in the memory of the computer. The CPU therefore reads an instruction and executes it and continues doing so until it reaches the end of the program. This stored program is called software in computer terminology. Since

it is the software, which runs any computer, the computer is as good as the program that is running at any given time.

CAD/CAM HARDWARE

The developments that have fuelled the growth of the CAD/CAM industry are mostly based on the major developments in microelectronics, which have helped in leapfrogging the capabilities.

The major component of any CAD/CAM system is the basic hardware that goes with the computational aspect, that is, the computer and the associated peripherals that go with it.

Basic Structure

The computing system in operation can be compared to a human being in to of its operating characteristics. The basic configuration of a typical computer system is shown in Fig. 1.7.

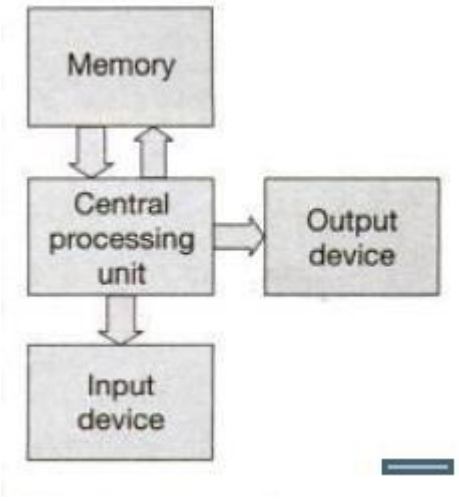


Fig 1.7: Configuration of a typical CAD system

The heart of any computing system is the Central Processing Unit (CPU). It is in the CPU that all the necessary functions of a computer are carried out. The main functions performed in the CPU are arithmetic and logic operations The CPU communicates with the external world through input/output devices or, in short, I/O devices. These are similar to the sensory organs by which a human being maintains contact with the outside environment. These are also collectively called peripheral units. Through an input device, the user would be able to communicate with the CPU, either to give certain data or to control the operation of the CPU. The output device is a means through which the CPU gives the results of the computations.

Another important unit of a computer system is the memory unit. These are the areas where the necessary data or program (sequence of instructions) is stored. The type of memory and its amount determines the capabilities of a typical computing computing system. The present-day computers work on a principle called the stored-program concept. It means that the sequence of operations to be carried out by the CPU is stored in the memory of the computer. The CPU therefore reads an instruction and executes it and continues doing so until it reaches the end of the program. This stored program is called software in computer terminology. Since

it is the software, which runs any computer, the computer is as good as the program that is running at any given time.

CAD/CAM HARDWARE

The developments that have fuelled the growth of the CAD/CAM industry are mostly based on the major developments in microelectronics, which have helped in leapfrogging the capabilities.

The major component of any CAD/CAM system is the basic hardware that goes with the computational aspect, that is, the computer and the associated peripherals that go with it.

Basic Structure

The computing system in operation can be compared to a human being in to of its operating characteristics. The basic configuration of a typical computer system is shown in Fig. 1.7.

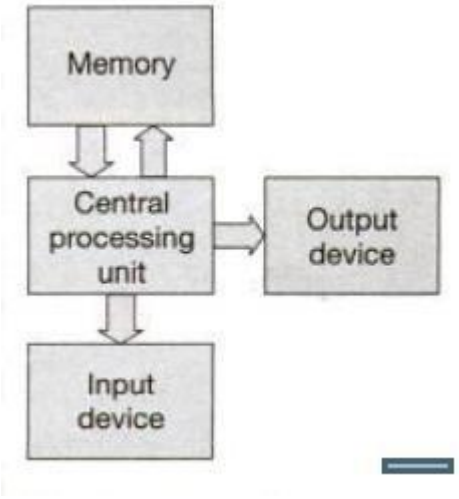


Fig 1.7: Configuration of a typical CAD system

The heart of any computing system is the Central Processing Unit (CPU). It is in the CPU that all the necessary functions of a computer are carried out. The main functions performed in the CPU are arithmetic and logic operations The CPU communicates with the external world through input/output devices or, in short, I/O devices. These are similar to the sensory organs by which a human being maintains contact with the outside environment. These are also collectively called peripheral units. Through an input device, the user would be able to communicate with the CPU, either to give certain data or to control the operation of the CPU. The output device is a means through which the CPU gives the results of the computations.

Another important unit of a computer system is the memory unit. These are the areas where the necessary data or program (sequence of instructions) is stored. The type of memory and its amount determines the capabilities of a typical computing computing system. The present-day computers work on a principle called the stored-program concept. It means that the sequence of operations to be carried out by the CPU is stored in the memory of the computer. The CPU therefore reads an instruction and executes it and continues doing so until it reaches the end of the program. This stored program is called software in computer terminology. It is

the software, which runs any computer, the computer is as good as the program that is running at anygiven time.

1. Work Station

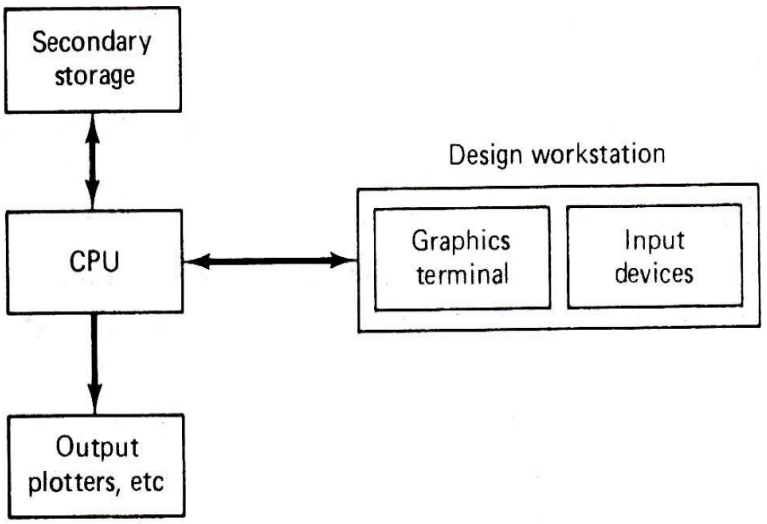


Fig. 1.9 Design Workstation

• The work station is a visible part of the CAD system which provides interaction between the operator and the system.

• Among these advantages offered by work station are their availability, portability, the availability to dedicate them to a single task without affecting other users and their consistency of time response.

• A work station can be defined as a station of work with its own computing power to support major software packages, multitasking capabilities demanded by increased usage, complex tasks and networking potential with other computing environments.

2. Input Devices

- A no. of input devices is available. These devices are used to input two possible types of information: text and graphics.
- Text-input devices and the alphanumeric keyboards.
- There are two classes of graphics input devices: Locating devices and image input devices.
- Locating devices, or locators, provide a position or location on the screen.
- These include light pens, mouse, digitizing tablets, joysticks, trackballs, thumbwheels, touchscreen and touchpads.

- Locating devices typically operate by controlling the position of a cursor on screen. Thus, they are also referred to as cursor-control devices.

I. Scanners

Scanners comprise other class of graphics-input device.

- There are four relevant parameters to measure the performance of graphics input devices. These are resolution, accuracy, repeatability and linearity.
- Some may be more significant to some devices than others.

II. Keyboards

- Keyboards are typically employed to create/edit programs or to perform word processing functions.
- CAD/CAM systems, information entered through keyboards should be displayed back to the user on a screen for verification.

III. Digitizing Tablets

- A digitizing tablet is considered to be a locating as well as pointing device. It is a small, low-resolution digitizing board often used in conjunction with a graphics display.
- The tablet is a flat surface over which a stylus can be moved by the user.
- A tablet's typical resolution is 200 dots per inch
- The tablet operation is based on sensitizing its surface area to be able to track the pointing element motion on the surface.
- Several sensing methods and technologies are used in tablets. The most common sensing technology is electromagnetic, where the pointing element generates an out of phase magnetic field sensed by wire grid in tablet surface.

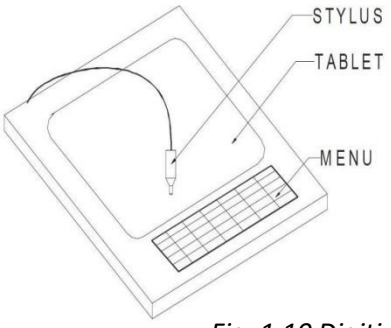


Fig. 1.10 Digitizer

IV. Mouse

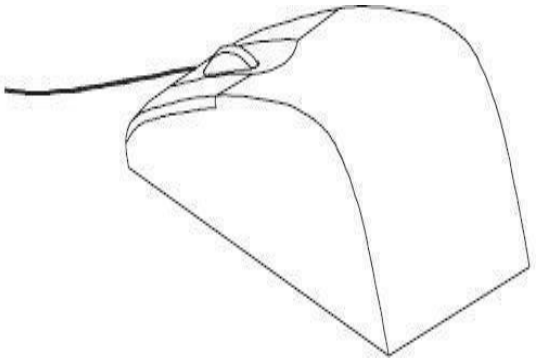


Fig. 1.11 Mouse

- There are two basic types of mouse available mechanical and optical.
- The mechanical mouse has roller in order to record the mouse motion in X and Y directions.
- In optical mouse, movements over the surface are measured by a light beam modulation technique.
- The light source is located at the bottom and the mouse must be in contact with the surface for screen cursor to follow its movement.

V. Joy sticks & Trackballs

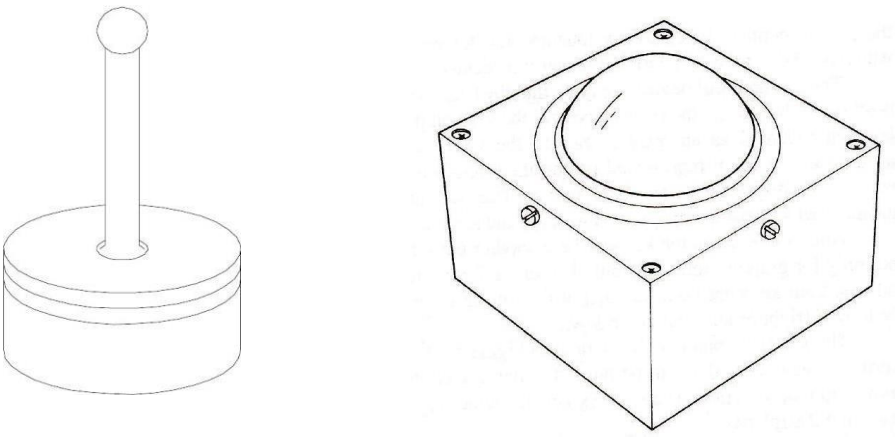


Fig. 1.10 Joy stick & Track ball

- The joystick works by pushing its stick backwards or forward or to left or right. The extreme positions of these directions correspond to the four corners of the screen.
- A trackball is similar in principal to a joystick but allows more precise fingertip control. The ball rotates freely within its mount.
- Both the joystick and trackball are used to navigate the screen display cursor. The user of a trackball can learn quickly how to adjust to any nonlinearity in its performance.

VI. Thumbwheels

- Two thumbwheels are usually required to control the screen cursor, one for its horizontal position and other for its vertical position. Each position is indicated onscreen by cross-hair.

COMPUTER GRAPHICS

Computer graphics is an art of drawing pictures on computer screens with the help of programming. It involves computations, creation, and manipulation of data. In other words, computer graphics is a rendering tool for the generation and manipulation of images.

• Computer is information processing machine. User needs to communicate with computer and the computer graphics is one of the most effective and commonly used ways of communication with theuser.

• It displays the information in the form of graphical objects such as pictures, charts, diagram andgraphs.

• Graphical objects convey more information in less time and easily understandable formats forexample statically graph shown in stock exchange.

• In computer graphics picture or graphics objects are presented as a collection of discrete pixels.

• We can control intensity and color of pixel which decide how picture look like.

• The special procedure determines which pixel will provide the best approximation to the desiredpicture or graphics object this process is known as Rasterization.

• The process of representing continuous picture or graphics object as a collection of discrete pixels is called Scan Conversion.

Advantages of computer graphics

• Computer graphics is one of the most effective and commonly used ways of communication withcomputer.

• It provides tools for producing picture of "real-world" as well as synthetic objects such as mathematical surfaces in 4D and of data that have no inherent geometry such as survey result.

• It has ability to show moving pictures thus possible to produce animations with computer graphics.

• With the use of computer graphics we can control the animation by adjusting the speed, portion of

picture in view the amount of detail shown and so on.

• It provides tools called motion dynamics. In which user can move objects as well as observes as perrequirement for example walk throw made by builder to show flat interior and surrounding.

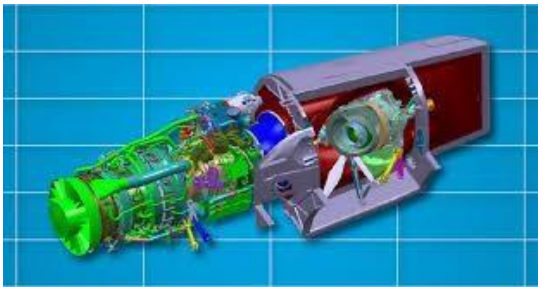
• It provides facility called update dynamics. With this we can change the shape color and otherproperties of object.

Applications of Computer Graphics

Computer Graphics has numerous applications, some of which are listed below:

- User interface: - Visual object which we observe on screen which communicates with user isone of the most useful applications of the computer graphics.
- Plotting of graphics and chart in industry, business, government and educational organizations drawing like bars, pie-charts, histogram's are very useful for quick and good decision making.
- Office automation and desktop publishing: - It is used for creation and dissemination of information. It is used in in-house creation and printing of documents which contains text, tables, graphs and other forms of drawn or scanned images or picture.
- Computer aided drafting and design: - It uses graphics to design components and system suchas automobile bodies structures of building etc.
- Simulationandanimation:- Useof graphics in simulationmakesmathematicmodelsand mechanical systems more realistic and easy to study.
- Artandcommerce:- Therearemanytoolsprovidedbygraphicswhichallowsusedtomaketheir picture animated and attracted which are used in advertising.
- Processcontrol:- Now aday's automation is usedwhich is graphically displayed on thescreen.
- Cartography: - Computer graphics is also used to represent geographic maps, weather maps,oceanographic charts etc.
- Education and training: - Computer graphics can be used to generate models of physical, financial and economic systems. These models can be used as educational aids.
- Image processing: - It is used to process image by changing property of the image.

Example of graphic design of a car



Output Devices

- CAD/CAM applications require output devices such as displays and hardcopy printers.

DISPLAY DEVICES

- Display devices are also known as output devices.
- Most commonly used output device in a graphics system is a video monitor.

Cathode Ray Tube

The primary output device in a graphical system is the video monitor. The main element of a video monitor is the Cathode Ray Tube (CRT), shown in the following illustration.

- It is an evacuated glass tube.
- An electron gun at the rear of the tube produce a beam of electrons which is directed towards thescreen of the tube by a high voltage typically 15000 to 20000 volts
- Innersidescreen is coatedwith phosphorsubstance whichgiveslightwhen it is strokedbyeelectrons.
- Control grid controls velocity of electrons before they hit the phosphor.
- The control grid voltage determines how many electrons are actually in the electron beam. The negative the control voltage is the fewer the electrons that pass through the grid.
- Thus control grid controls Intensity of the spot where beam strikes the screen.
- The focusing system concentrates the electron beam so it convergesto small point when hits thephosphor coating.
- Deflection system directs beam which decides the point where beam strikes the screen.
- Deflectionsystem of the CRTconsists of two pairs of parallel plates whicharevertical and horizontaldeflection plates.
- Voltage applied tovertical andhorizontal deflection plates is controlvertical and horizontal

deflection respectively.

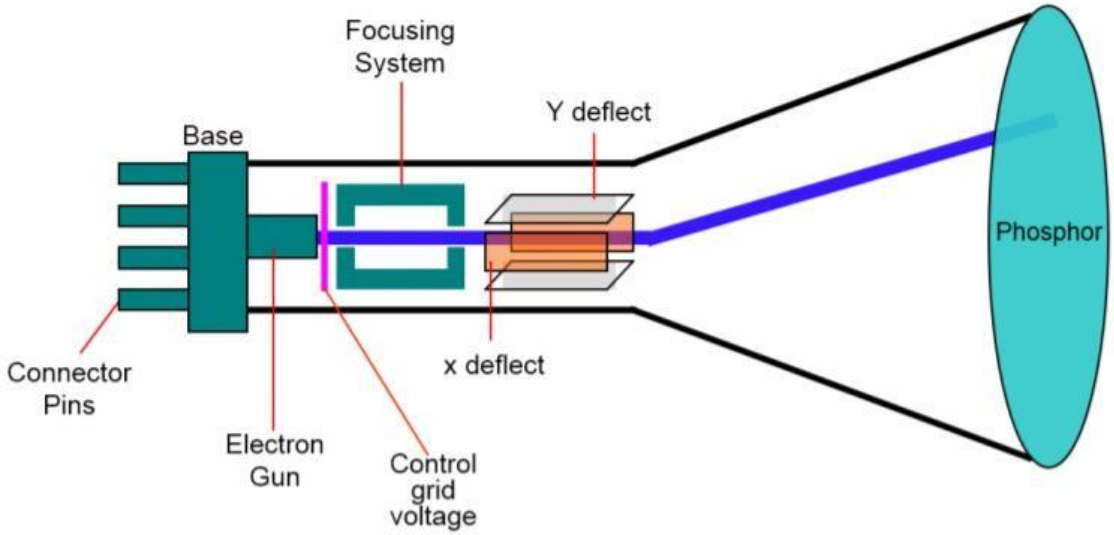


Fig 1.11: Cathode Ray Tube

There are two techniques used for producing images on the CRT screen:

1. Vector scan/Random scan display.
2. Raster scan display.

Vector scan/Random scan display

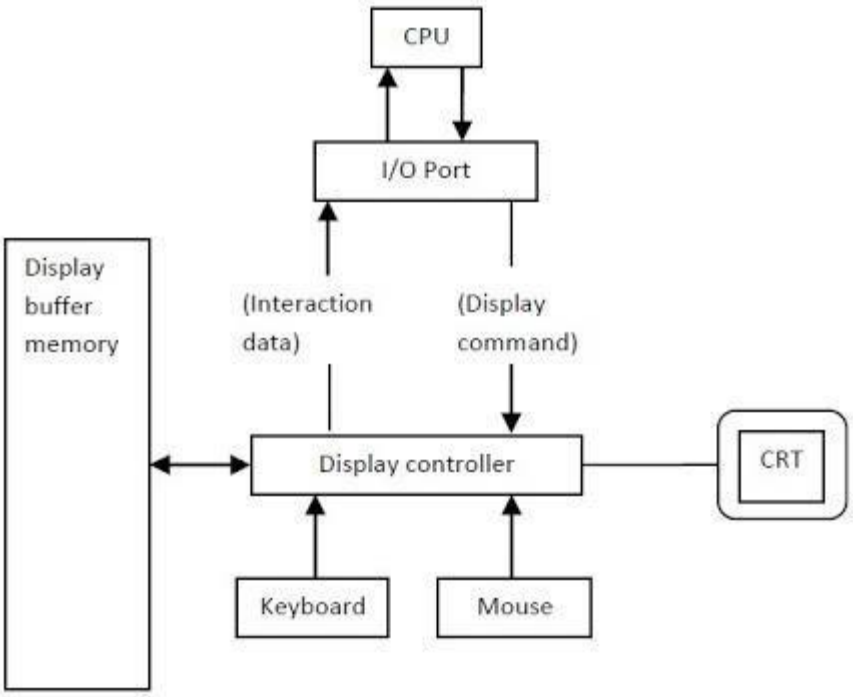


Fig 1.12: Architecture of a Vector Scan Display

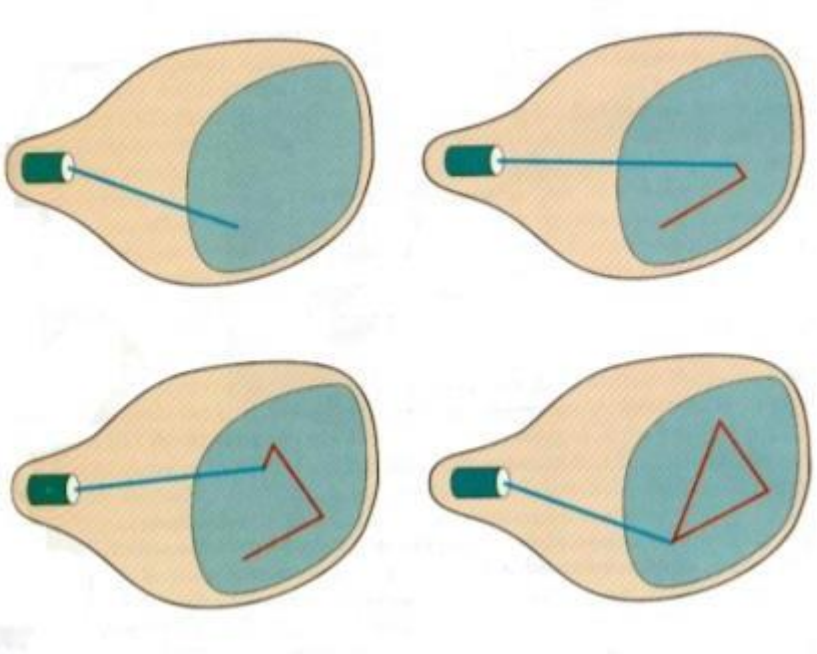


Fig 1.12: Vector Scan/ Raster Scan Display

- Random scan also is known as a vector display.
- Random scan display directly traces out only the desired lines on CRT.
- If we want line between point p1 & p2 then we directly drive the beam deflection circuitry which focusbeam directly from point p1 to p2
- If we do not want to display line from p1 to p2 and just move then we can blank the beam as we moveit.
- To move the beam across the CRT, the information about both magnitude and direction is required. Thisinformation is generated with the help of vector graphics generator.
- Image shows the architecture of vector display. It consists of display controller, CPU, display buffermemory and CRT.
- Display controller is connected as an I/O peripheral to the CPU.
- Display buffer stores computer produced display list or display program.
- The Program contains point & line plotting commands with end point co-ordinates as well as characterplotting commands.
- Display controller interprets command and sends digital and point co-ordinates to a vector generator.
- Vector generator then converts the digital co-ordinate value to analog voltages for beam deflectioncircuits that displace an electron beam which points on the CRT's screen.
- In this technique beam is deflected from end point to end point hence this techniques is also calledrandom scan.

- We know as beam strikes phosphors coated screen it emits light but that light decays after few milliseconds and therefore it is necessary to repeat through the display list to refresh the screen at least30 times per second to avoid flicker.
- As display buffer is used to store display list and used to refreshing, it is also called refresh buffer.

Raster Scan

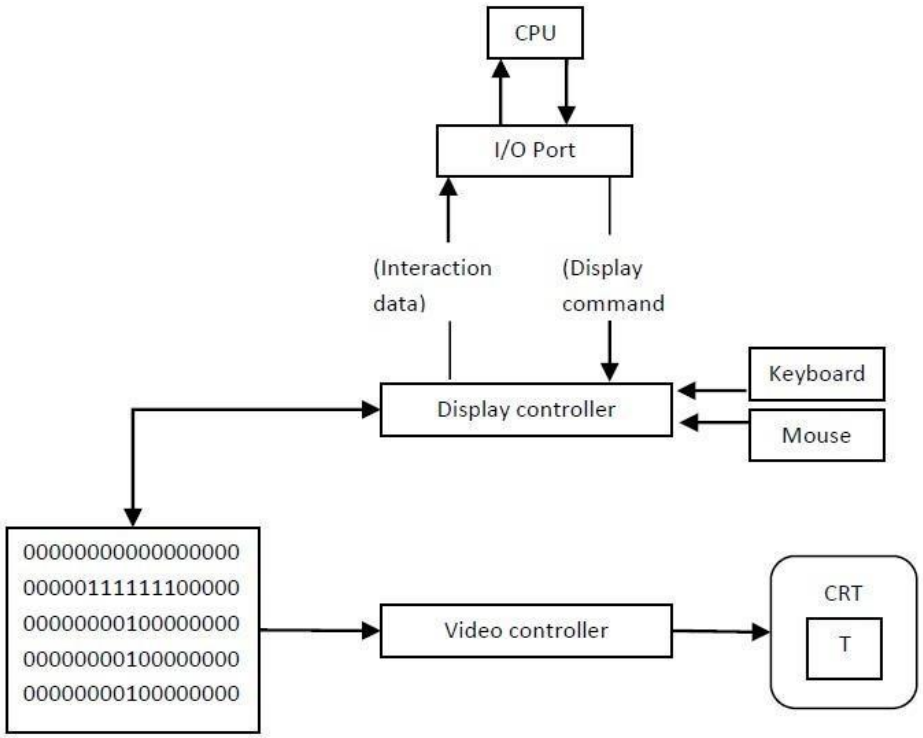


Fig 1.13: Architecture of Raster Display

- It consists of a display controller, CPU, video controller, refreshes buffer, keyboard, mouse, and CRT.
- The display image is stored in the form of 1's and 0's in the refresh buffer.
- The video controller reads this refresh buffer and produces the actual image on screen.
- It will scan one line at a time from top to bottom & then back to the top.

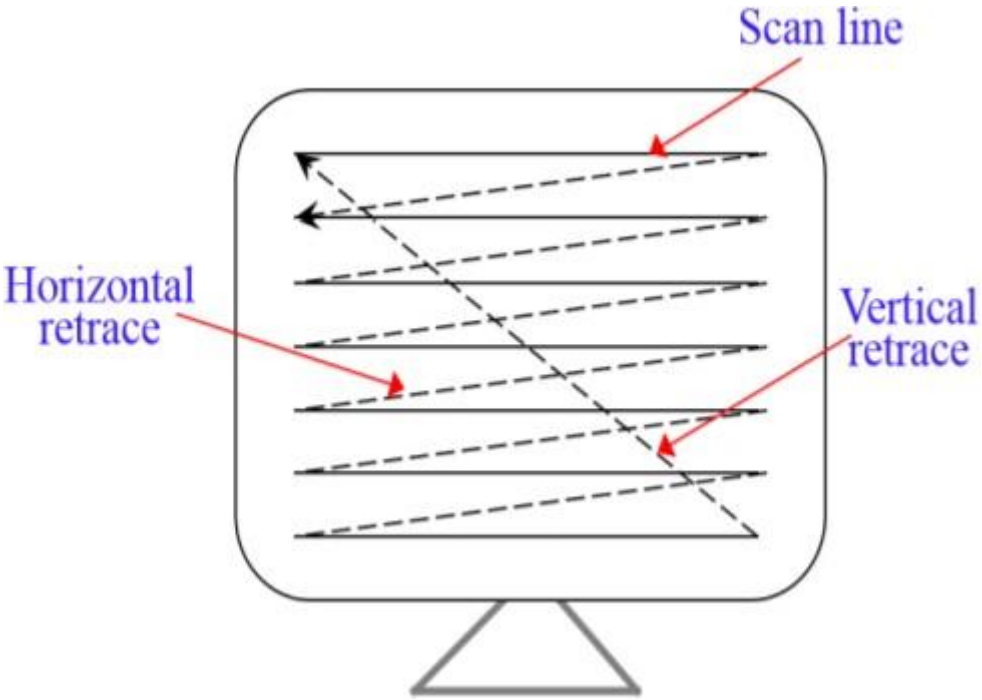


Fig 1.14: Raster scan CRT

- In this method the horizontal and vertical deflection signals are generated to move the beam allover the screen in a pattern shown in Image.
- Here beam is swept back & forth from left to the right.
- When beam is moved from left to right it is ON.
- When beam is moved from right to left it is OFF and process of moving beam from right to left aftercompletion of row is known as Horizontal Retrace.
- When beam is reach at the bottom of the screen. It is made OFF and rapidly retraced back to the topleft to start again and process of moving back to top is known as Vertical Retrace.
- The screen image is maintained by repeatedly scanning the same image. This process is known asRefreshing of Screen.
- In raster scan displays a special area of memory is dedicated to graphics This memory is calledFrame Buffer.
- Frame buffer holds set of intensity values for all the screen points
- That intensity is retrieved from frame buffer and display on screen one row at a time.
- Each screen point referred as pixel or Pel (Picture Element).
- Each pixel can be specified by its row and column Numbers.

Difference between random scan and raster scan

Base of Difference	Raster Scan System	Random Scan System
Electron Beam	The electron beam is swept across the screen, one row at a time, from top to bottom.	The electron beam is directed only to the parts of screen where a picture is to be drawn.
Resolution	Its resolution is poor because raster system in contrast produces zigzag lines that are plotted as discrete point sets.	Its resolution is good because this system produces smooth lines drawings because CRT beam directly follows the line path.
Picture Definition	Picture definition is stored as a set of intensity values for all screen points, called pixels in a refresh buffer area.	Picture definition is stored as a set of line drawing instructions in a display file.
Realistic Display	The capability of this system to store intensity values for pixel makes it well suited for the realistic display of scenes contain shadow and color pattern.	These systems are designed for line-drawing and can't display realistic shaded scenes.
Draw an Image	Screen points/pixels are used to draw an image.	Mathematical functions are used to draw an image.

Direct-view storage tubes (DVST)

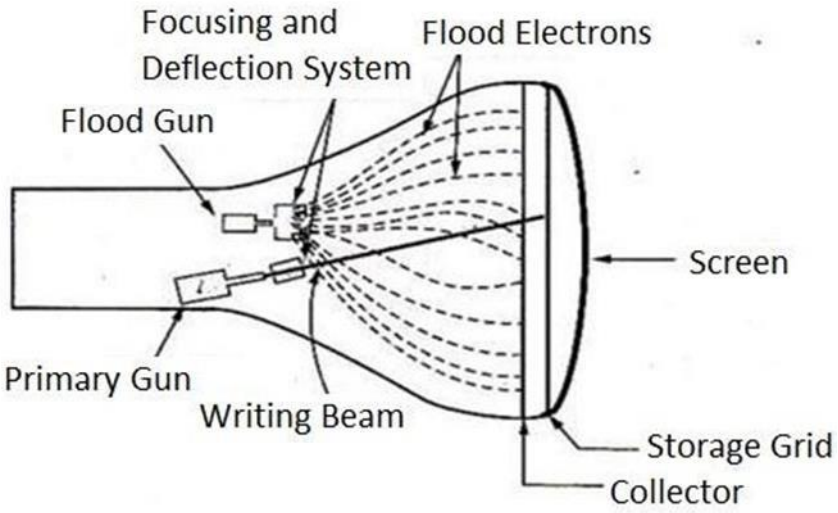


Fig 1.15: Direct-view storage tube

- In raster scan display we do refreshing of the screen to maintain a screen image.
- DVST gives alternative method for maintaining the screen image.
- DVSTusesthestoragegridwhichstoresthepictureinformationasachargedistributionjustbehindthe phosphor coated screen.
- DVST consists two electron guns a primary gun and a flood gun.
- A primary gun stores the picture pattern and the floodgun maintains the picture display.
- Aprimary gunemits highspeedelectrons whichstrike on thestorage gridto draw thepicture pattern.
- Aselectronbeamstrikes on the storage gridwithhighspeed, it knocksoutelectronsfromthe storage grid keeping the net positive charge.
- The knocked out electrons are attracted towards the collector.
- The net positive charge on the storage grid is nothing but the picture pattern.
- Thecontinuous low speedelectronsfromfloodgun pass throughthe control grid and are attracted to the positive charged area of the storage grid.
- The low speed electrons then penetrate the storage grid and strike the phosphor coating without affecting the positive charge pattern on the storage grid.
- During this process thecollector just behindthe storage gridsmoothout theflow of floodelectrons.

Advantage of DVST

- Refreshing of CRT is not required.
- Very complex pictures can be displayed at very high resolution without flicker.
- Flat screen.

Disadvantage of DVST

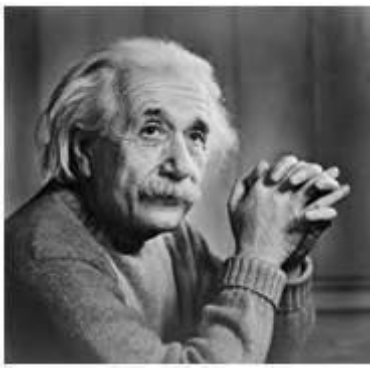
- They do not display color and are available with single level of line intensity.
- For erasing it is necessary toremoval ofcharge on thestorage gridsoerasing and redrawing process take several second.
- Erasing selective part of the screen cannot be possible.
- Cannot used for dynamic graphics application as on erasing it produce unpleasant flash over entire screen.
- It has poor contrast as a result of the comparatively low accelerating potential applied to the flood electrons.
- The performance of DVST is somewhat inferior to the refresh CRT.

Pixel

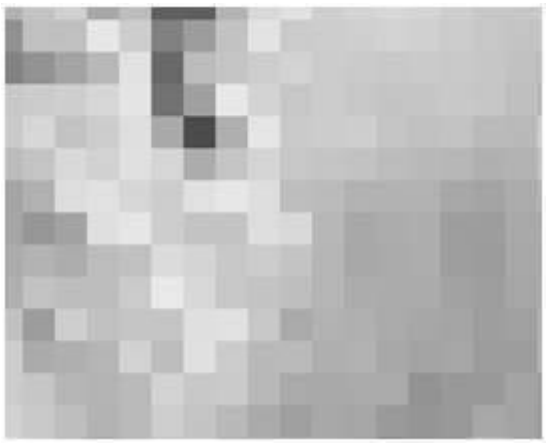
Pixel is the smallest element of an image. Each pixel correspond to any one value. In an 8-bit gray scale image, the value of the pixel between 0 and 255. The value of a pixel at any point correspond to the intensity of the light photons striking at that point. Each pixel store a value proportional to the light intensity at that particular location.

PEL

A pixel is also known as PEL. You can have more understanding of the pixel from the pictures given below.



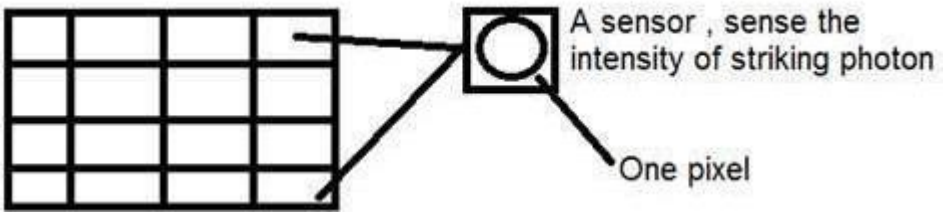
In the above picture, there may be thousands of pixels, that together make up this image. We will zoom that image to the extent that we are able to see some pixels division. It is shown in the image below.



Relation ship with CCD (Charge - Coupled Device) array

We have seen that how an image is formed in the CCD array. So a pixel can also be defined as The smallest division the CCD array is also known as pixel.

Each division of CCD array contains the value against the intensity of the photon striking to it. This value can also be called as a pixel.



Calculation of total number of pixels

We have define an image as a two dimensional signal or matrix. Then in that case the number of PEL would be equal to the number of rows multiply with number of columns.

This can be mathematically represented as below:

Total number of pixels = number of rows (X) number of columns

Or we can say that the number of (x,y) coordinate pairs make up the total number of pixels.

We will look in more detail in the tutorial of image types , that how do we calculate the pixels in a color image.

Gray level

The value of the pixel at any point denotes the intensity of image at that location , and that is also known as gray level.

Pixel value.(0)

Each pixel can have only one value and each value denotes the intensity of light at that point of the image.

We will now look at a very unique value 0. The value 0 means absence of light. It means that 0 denotes dark, and it further means that when ever a pixel has a value of 0, it means at that point, black color would be formed.

Have a look at this image matrix

0	0	0
0	0	0
0	0	0

Now this image matrix has all filled up with 0. All the pixels have a value of 0. If we were to calculate the total number of pixels form this matrix , this is how we are going to do it.

Total no of pixels = total no. of rows X total no. of columns

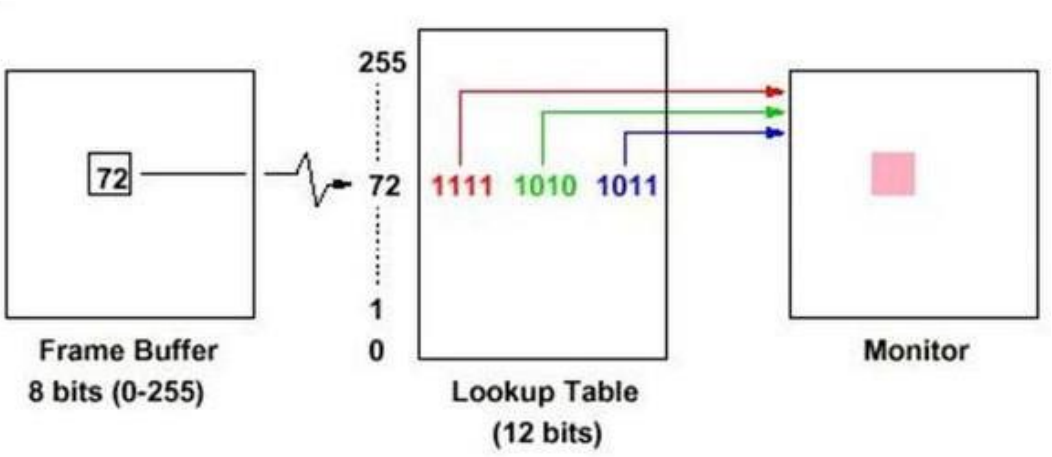
= 3 X 3 = 9.

It means that an image would be formed with 9 pixels, and that image would have a dimension of 3 rows and 3 column and most importantly that image would be black.

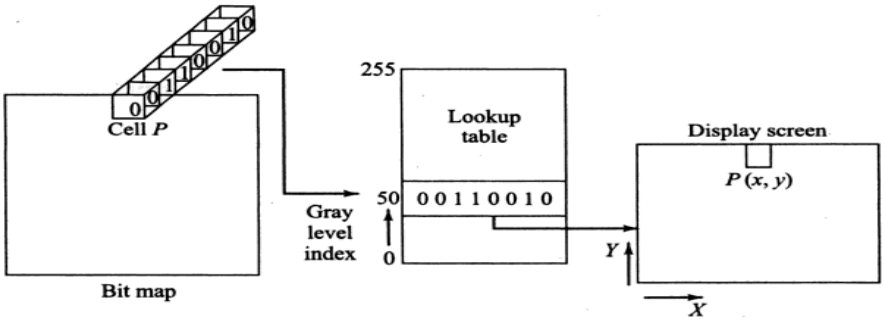
The resulting image that would be made would be something like this

Lookup Tables

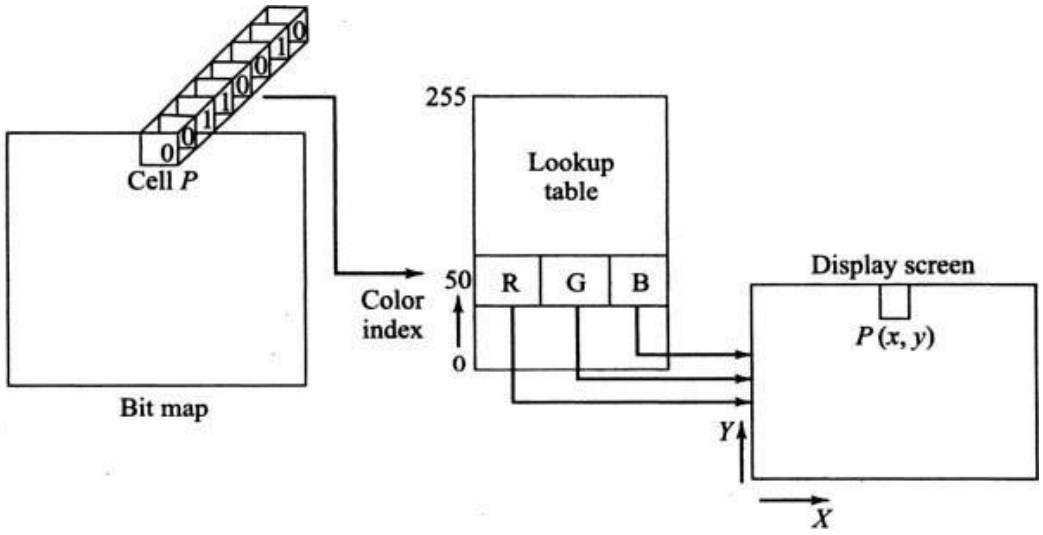
Video controller often uses a lookup table to allow indirection of display values in frame buffer.



- Allows flexible use of colors without lots of frame-buffer memory.
- Allows change of display without remapping underlying data double buffering.
- Permits simple animation.
- Common sizes: 8 x 12; 8 x 24; 12 x 24.



(a) Monochrome display



(b) Color display

Fig. 1.16 Relationship between pixel value and a lookup table

Frame Buffer

The frame buffer is the video memory that is used to hold or map the image displayed on the screen.

The amount of memory required to hold the image depend primarily on the resolution of the screen image andthe colour depth.

The formula to calculate how much video memory is required at a given resolution and bit depth is givenbelow.

- Memory in MB = (X-resolution*Y-resolution*Bit per pixel)/(8*1024* 1024)

Liquid Crystal Display (LCD)

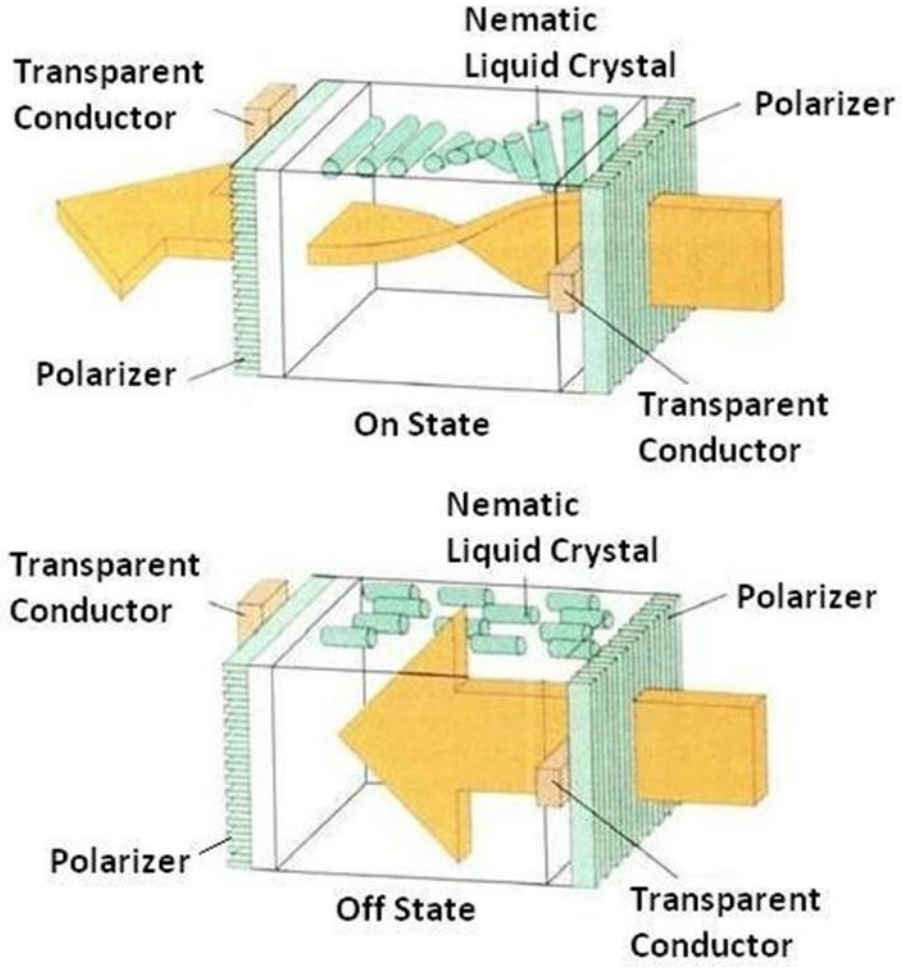


Fig 1.17: Light twisting shutter effect used in design of most LCD.

- It is generally used in small system such as calculator and portable laptop.
- This non emissive device produce picture by passing polarized light from the surrounding or from an internal light source through liquid crystal material that can be aligned to either block or transmit the light.
- The liquid crystal refreshes to fact that these compounds have crystalline arrangement of molecules then also flows like liquid.
- It consists of twoglass plates eachwithlight polarizer at right angles toeachothersandwich the liquid crystal material between the plates.
- Rows of horizontal transparent conductors are built into one glass plate, and column of vertical conductors are put into the other plates.
- The intersection of two conductors defines a pixel position.
- In the ON state polarized light passing through material is twisted so that it will pass through the opposite polarizer.
- In the OFF state it will reflect back towards source.
- We applied a voltage to the two intersecting conductor to align the molecules so that the light is

not twisted.

- This type of flat panel device is referred to as a passive matrix LCD.
- In active matrix LCD transistors are used at each (x, y) grid point.

  Transistorcausecrystalto changetheirstatequicklyandalsoto controldegreetowhichthestatehas been changed.
- Transistor can also serve as a memory for the state until it is changed.
- So transistor make cell ON for all time giving brighter display then it would be if it had to be refreshperiodically

Advantages of LCD display

- Low cost.
- Low weight.
- Small size
- Low powerconsumption.

LED (Light Emitting Diode)

A Light Emitting Diode is a semiconductor device that emits visible light of a certain color, and is fundamentally different from conventional light sources such as incandescent, fluorescent, and gas - discharge lamps, in that an LED:

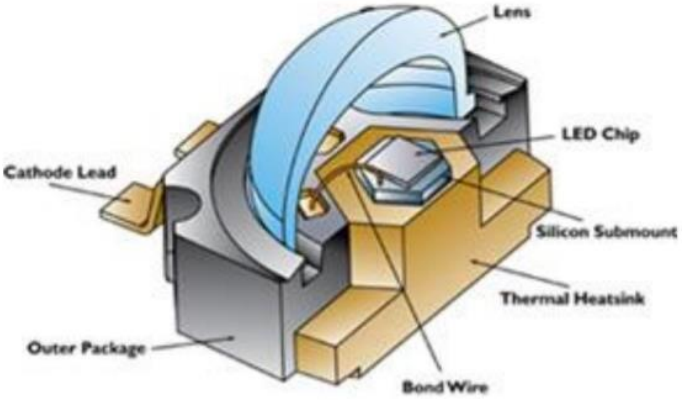
- uses no gas or filament, has no glass bulb,
- and no failure -prone moving parts

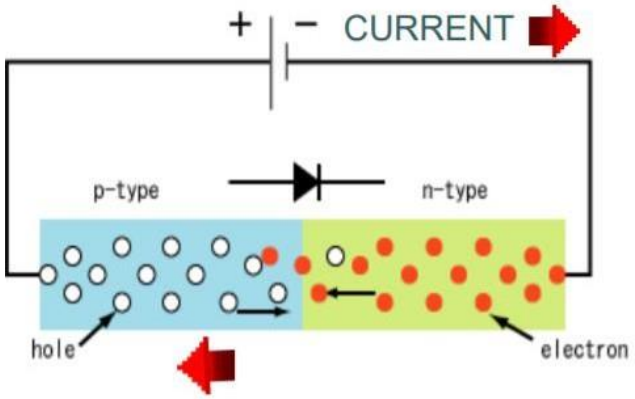
➢ Like a normal diode, the LED consists of a chip of semiconducting material impregnated, or doped withimpurities to create a p -n, (positive / negative) junction.

➢ Atoms in the n -type material have extra electrons, atoms in the p - type material have electron holes.

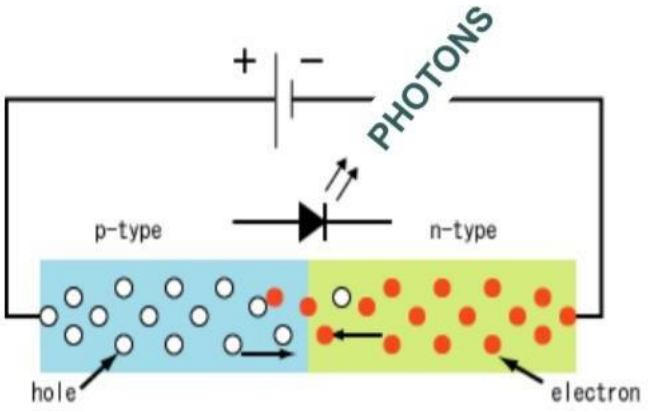
➢ Applying current pushes the atoms toward the junction. When they get close, the n -type atoms 'donate'their extra electrons to the p -type atoms which 'accept' them.

➢ A negative charge to the n - side allows current to flow from the (-) charged area to the (+) chargedarea. This is called 'forward bias'.





- When extra electrons in the n-type material fall into the holes in the p-type material, they release energy in the form of photons. The material in an LED is selected so that the wavelength of the photonsfalls within the visible portion of spectrum.
    - Different materials produce photons at different wavelengths / color.



Advantages

- Comparable in efficacy to CFLs, gaining on fluorescent tubes, and incandescents.
- Fixtures are directional, allowing for more efficient optics.

- Quality of White Light LEDs now comparable to CFLs, recent advances assure betterconsistency in color and CCT.
- Significantly longer 'Useful' life
- Light output has improved by 35% / year, while cost has dropped by 20% / year No infrared,(IR) radiation
- No ultraviolet, (UV) rays
- Mercury free
- Can operate in cold environments
- Can withstand impact and vibrations
- Inherently digital for ease of control
- Instant on
- Growing trend to modularity

CONCEPT OF COORDINATE SYSTEM

A coordinate system is a way of assigning numbers to points. In two dimensions, you need a pair of numbers to specify a point. The coordinates are often referred to as x and y, although of course, the names are arbitrary.
More than that, the assignment of pairs of numbers to points is itself arbitrary to a large extent. Points and objects are real things, but coordinates are just numbers that we assign to them so that we can refer to themeasily and work with them mathematically. We have seen the power of this when we discussed transforms, which are defined mathematically in terms of coordinates but which have real, useful physical meanings.

In a 2-D coordinate system the X axis generally points from left to right, and the Y axis generally points frombottom to top. (Although some windowing systems will have their Y coordinates going from top to bottom)

In three dimensions, you need three numbers to specify a point. (That's essentially what it means to be three dimensional.) The third coordinate is often called z. The z-axis is perpendicular to both the x-axis and the y-axis.

This demo illustrates a 3D coordinate system. The positive directions of the x, y, and z axes are shown as bigarrows. The x-axis is green, the y-axis is blue, and the z-axis is red.

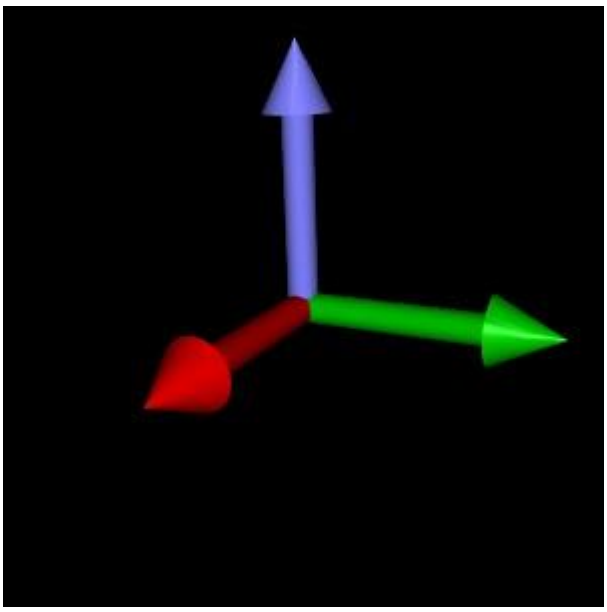
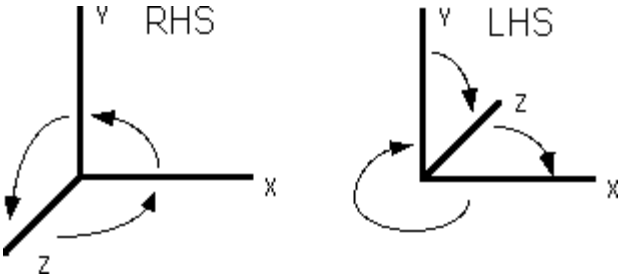


Fig 1.18: 3D Axes

When we add the third coordinate, Z, we have a choice as to whether the Z-axis points into the screen or out ofthe screen:



Right Hand Coordinate
System (RHS)Z is coming out
of the page

Counterclockwise rotations are positive
if we rotate about the X axis : the rotation Y->Z is positiveif we rotate about the Y axis : the rotation Z->X is positive if we rotate about the Z axis : the rotation X->Y is positive

Left Hand Coordinate
System (LHS)Z is going into
the page

Clockwise rotations are positive
if we rotate about the X axis : the rotation Y->Z is positive

f we rotate about the Y axis : the rotation Z->X is positiveif we rotate about the Z axis : the rotation X->Y is positive

Coordinate System

1. The Model Coordinate System or (world coordinate systems) (MCS).

2. The Working Coordinate System (WCS) or User Coordinate System (UCS).

3. The Screed Coordinate System (or device coordinate system) (SCS).

Model Coordinate System

The model coordinate system is defined as the reference space of the model with respect to which all themodel geometrical data is stored.

It is a cartesian system which forms the default coordinate system used by a particular software program.

The X, Y, and Z axes of the MCS can be displayed on the computer screen. The origin of the MCS can be arbitrarily chosen by the user while its orientation is established by the software.

The three default sketch planes of a CAD/CAM system define the three planes of the MCS, and their intersection point is the MCS origin. When a CAD designer begins sketching, the origin becomes a corner pointof the profile being sketched. The sketch plane defines the orientation of the profile in the model 3D space.
This is how we attach the MCS to a geometric model. In order for the user to communicate properly and effectively with a model database, the relationships between the MCS orthogonal (sketch) planes and themodel views must be understood by the user.

Typically, the software chooses one of two possible orientations of the MCS in space. As shown in Figure 1.18a,the XY plane is the horizontal plane and defines the model top view. The front and right side views are consequently defined by the XZ and YZ planes, respectively.

Figure 1.18b shows the other possible orientation of the MCS where the XY plane is vertical and defines the model front view. As a result, the XZ and the YZ planes define the top and the right side views, respectively.

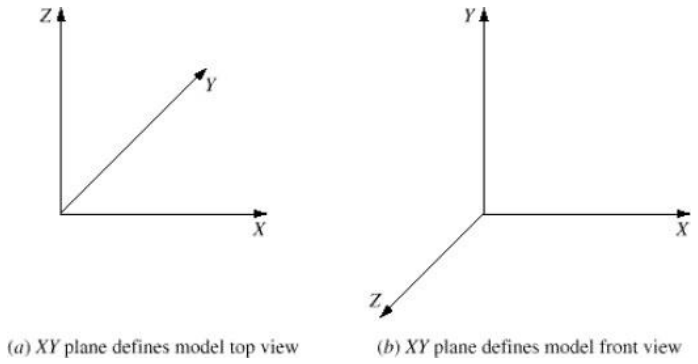


Fig 1.19: Possible orientations of MCS in 3D Space

Existing CAD/CAM software uses the MCS as the default WCS (see Section 2.4.2). In both orientations, the XY plane is the default construction (sketch) plane. If the user utilizes such a plane, the first face of a model to beconstructed becomes the top or front view, depending on which MCS is used.

The MCS is the only coordinate system that the software recognizes when storing or retrieving graphical information in or from a model database. Many existing software packages allow the user to input coordinateinformation in cartesian (x, y, z) and cylindrical (r,θ,z).

This input information is transformed to (x, y, z) coordinates relative to the MCS before being stored in the database. Obtaining views is a form of retrieving graphical information relative to the MCS. If the MCS orientation does not match the desired orientation of the object being modeled, users become puzzled and confused. Another form of retrieving information is entity verification. Coordinates of points defining the entityare given relative to MCS by default. However, existing software allows users to obtain the coordinates relativeto another system (WCS) by using the proper commands or modifiers.

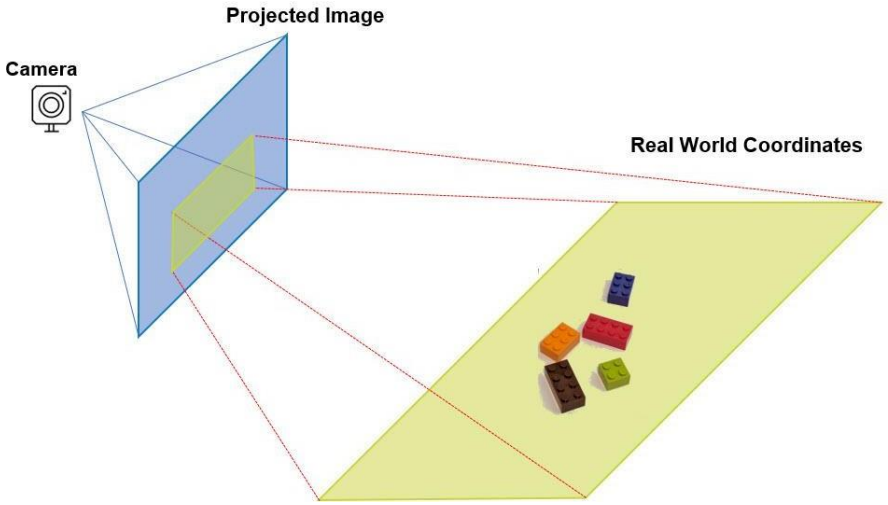


Fig 1.20: Model or World Coordinate System

Working Coordinate System

It is often convenient in the development of geometric models and the input of geometric data to refer to anauxiliary coordinate system instead of the MCS. This is usually useful when a desired plane (face) of construction is not easily defined as one of the MCS orthogonal planes, as in the case of inclined faces of a model.

The user can define a cartesian coordinate system whose XY plane is coincident with the desired plane of construction. That system is the Working Coordinate System, WCS. It is a convenient user-defined system thatfacilitates geometric construction.

It can be established at any position and orientation in space that the user desires. While the user can inputdata in reference to the WCS, the CAD software performs the necessary transformations to the MCS beforestoring the data. The ability to use two separate coordinate systems within the same model database in relation to one another gives the user great flexibility. Some commercial software refers to the WCS as is; Unigraphics offers an example. Other software refers to it as a sketch plane (Pro/E and SolidWorks) or construction plane.

A WCS requires three noncollinear points to define its XY plane. The first defines the ori-gin, the first and the second define the X axis, and the third point with the first define the Y axis. The Z axis is determined as the cross product of the two unit vectors in the directions defined by the lines connecting the first and the second (the X axis), and the first and the third points (Y axis). We will use the subscript w to distinguish the WCS axes from those of the MCS. The Xm,Y,„ plane becomes the active sketch (working) or construction plane if the user defines a WCS. In this case, the WCS and its corresponding XwY,„ plane override the MCS and the default sketch plane, respectively. As a matter of fact, the MCS with its default sketch plane is the default WCS with its X„,Y„, plane. All CAD/CAM software packages provide users with three standard WCSs (sketch planes) that correspond to the three standard views: Front, Top, and Right sides. The user can define other WCSs or sketchplanes.

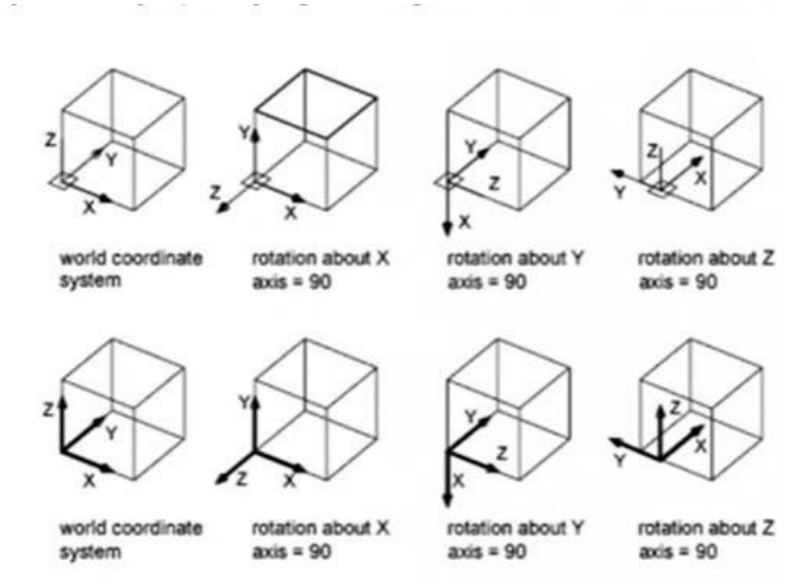


Fig 1.21: Working or User Coordinate System

Screen Coordinate System

In contrast to the MCS and WCS, the screen coordinate system (SCS) is defined as a 2D device-dependent coordinate system whose origin is usually located at the lower left corner of the graphics display, as shown in Figure 2.6. The physical dimensions of a device screen (aspect ratio) and the type of device (raster) determine the range of the SCS. The SCS is mostly used in view-related clicks such as definitions of view origin and windowor clicking a view to select it for graphics operations.

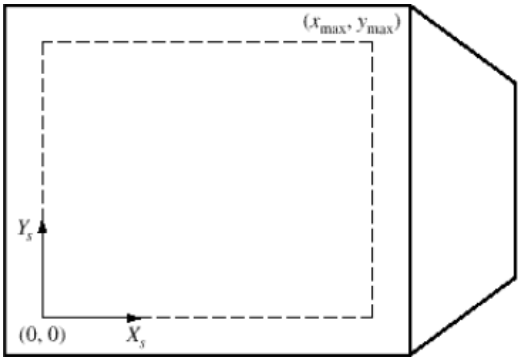


Fig 1.21: Typical SCS

- 2D Regular Cartesian Grid
- Origin (0,0) at lower left corner
- Horizontal axis – x
- Vertical axis – y
- Pixels are defined at the grid intersections
- This coordinate system is defined relative to the display window origin

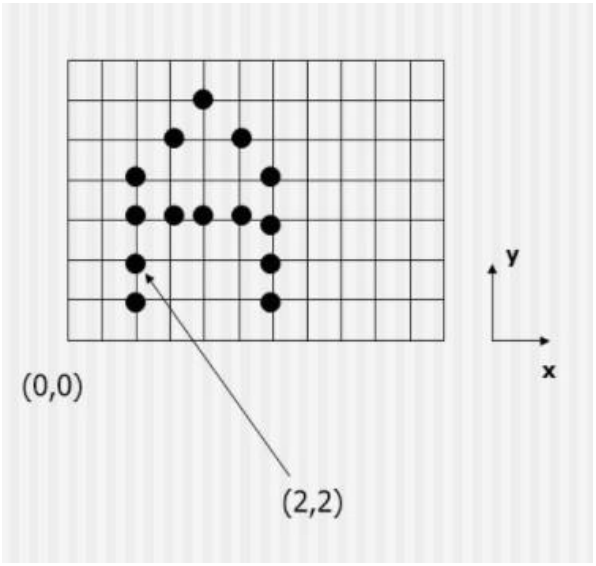
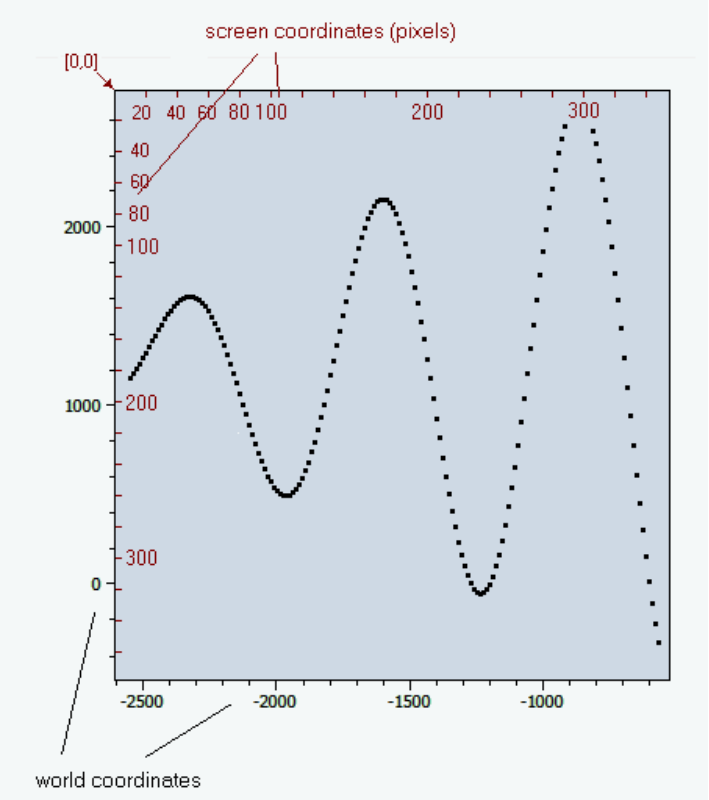


Fig 1.12: Screen Coordinate System

A 1024 1024 display has an SCS with a range of (0,0) to (1024,1024). The center of the screen has coordinates of (512,512). This SCS is used by the CAD/CAM software to display rel-evant graphics by converting directly from MCS coordinates to SCS (physical device) coordinates. A normalized SCS can also be utilized. The range of the SCS can be chosen from (0,0) to (1,1). Such representation can be translated by device-dependent codes tothe appropriate physical device coordinates. The third method of defining an SCS is by using the drawing size that the user chooses. If a size A drawing is chosen, the range of the SCS becomes (0,0) to (11,8.5) while size B produces the range (0,0) to (17,11). The rationale behind this method stems from the conventional drawing board so that the drafting paper is represented by the device screen.

A transformation operation from MCS coordinates to SCS coordinates is performed by the software before displaying the model views and graphics. Typically, for a geometric model, there is a data structure to store itsgeometric data (relative to MCS), and a display file to store its display data (relative to SCS).



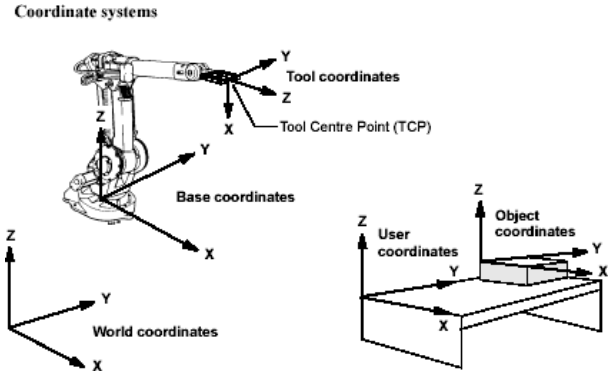


Fig 1.23: Coordinate Systems

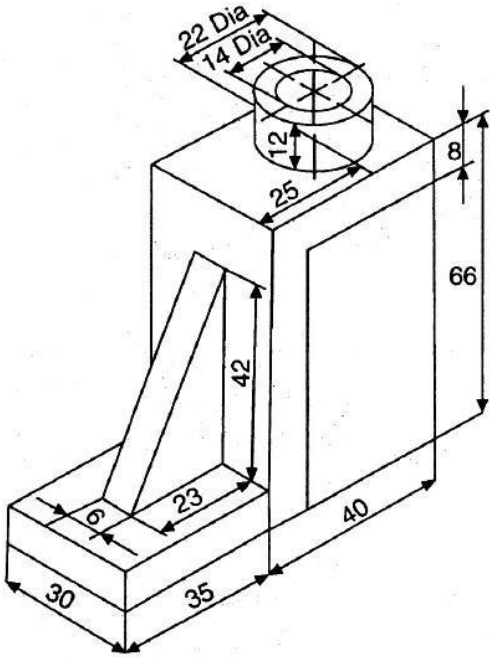


Fig. 1.24 Typical component to model.

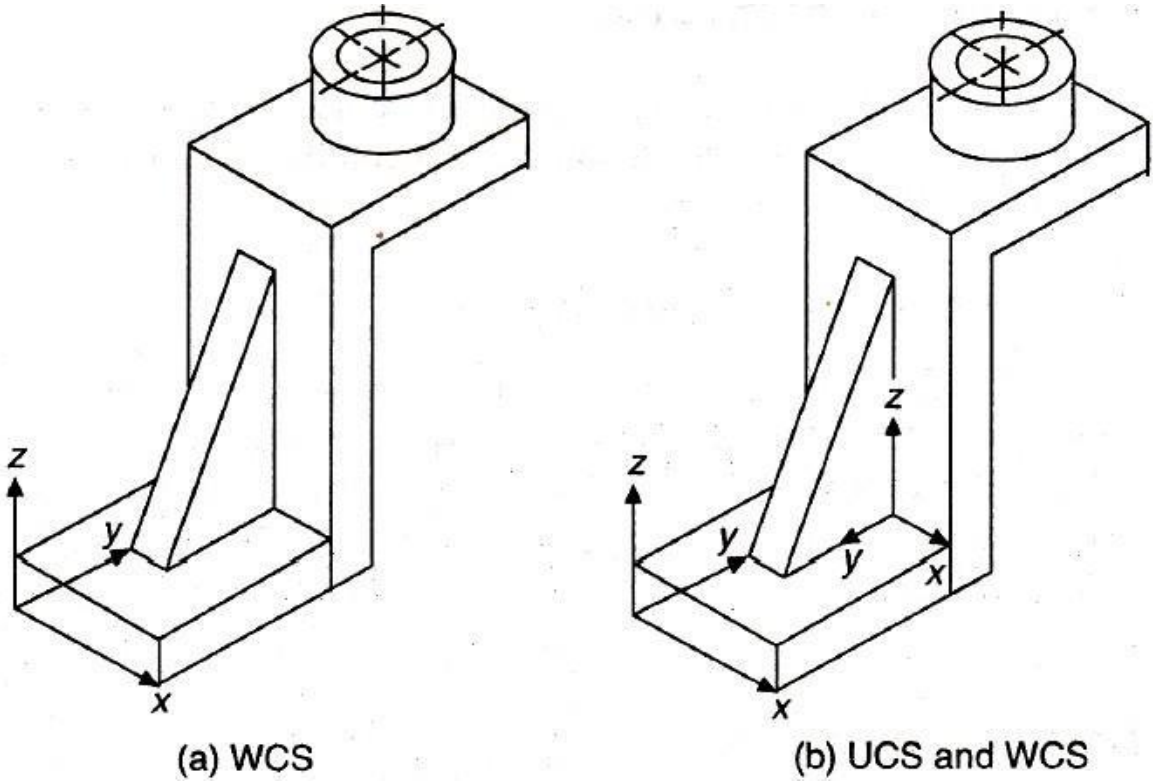


Fig. 1.25 Coordinate systems.

LINE GENERATION ALGORITHM

A line connects two points. It is a basic element in graphics. To draw a line, you need two points between whichyou can draw a line. In the following three algorithms, we refer the one point of line as X0, Y0 and the second point of line as X_1, Y_1.

- The Cartesian slop-intercept equation for a straight line is "$y = yy + y$" with 'y' representing slop and 'y' as theintercept.
- The two endpoints of the line are given which are say $(y1, y1)$ and $(y2, y2)$.

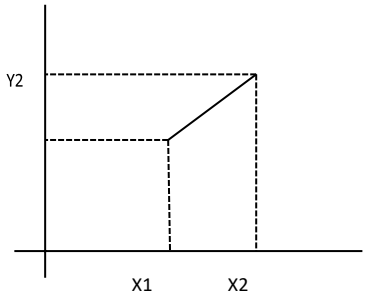


Fig. 1.26: - Line path between endpoint positions.

- We can determine values for the slope m by equation:

$y = (y2 - y1)/(y2 - y1)$

- We can determine values for the intercept b by equation:

$y = y1 - y * y1$

- For thegiveninterval Δy along aline, wecancompute thecorresponding$y$interval Δy as:

$\Delta y = y * \Delta y$

- Similarly for Δy:

$\Delta y = \Delta y/y$

- For line with slop $|y| < 1$, Δy can be set proportional to small horizontal deflection voltage and the corresponding verticaldeflectionvoltageisthensetproportionaltoΔywhichiscalculatedfromabove equation.
- For line with slop $|y| > 1$, Δy can be set proportional to small vertical deflection voltage and the corresponding horizontal deflection voltage is then set proportional to Δy which is calculated from above equation.
- Forline withslop$y = 1, \Delta y = \Delta y$ and thehorizontal and vertical deflectionvoltages are equal.

DDA Algorithm

Digital differential analyzer (DDA) is scan conversionline drawing algorithm based on calculating either Δy or Δx using above equation.

Step 1: Get the input of two end points (X_0, Y_0) and (X_1, Y_1). Step 2: Calculate the difference between two end points.

```
d
x
=
X1
-
X0
d
y
=
Y1
-
Y0
```

p 3: Based on the calculated difference in step-2, you need to identify the number of steps to put pixel.If dx > dy, then you need more steps in x coordinate; otherwise in y coordinate.

```
if(absolute(dx) >
absolute(dy))Steps =
absolute(dx);
else
Steps = absolute(dy);
```

Step 4: Calculate the increment in x coordinate and y coordinate.

```
Xincrement = dx /
    (float) steps;
Yincrement = dy /
    (float) steps;
```

Step 5: Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing ofthe line.

```
for(int v=0; v < Steps; v++)
{
x = x +
Xincreme
nt;y = y +
```

```
Yincreme
nt;
putpixel (Round(x), Round(y));
}
```

Advantages of DDA algorithm

- It is fasteralgorithm.
- It is simplealgorithm.

Disadvantage of DDA algorithm

- Floating point arithmetic is time consuming.
- Poor end pointaccuracy.

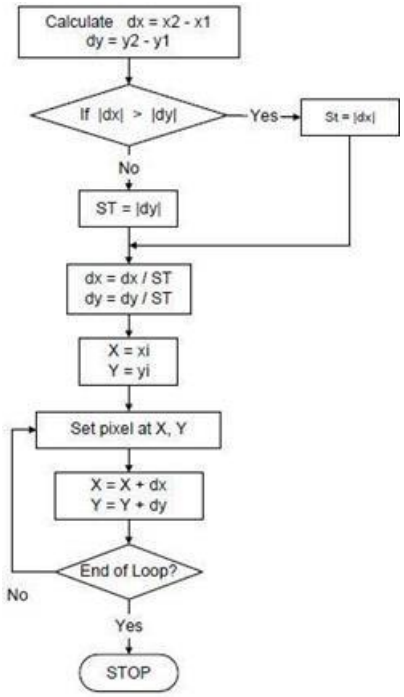


Fig. 1.27 Flow chart for line calculation procedure by DDA algorithm

Example 1.1

Indicate which raster locations would be chosen by DDA algorithm when scan converting a line from screen coordinate (19, 38) to screen coordinate (38, 52).

$$\mathbf{m} = \frac{\Delta y}{\Delta x} = \frac{y2-y1}{x2-x1} = \frac{52-38}{38-19} = 0.737$$

So, $m \leq 1$

ΔX= 1 and $y_{i+1} = y_i + m$

Table 1.1 Calculation of pixel positions by DDA Algorithm

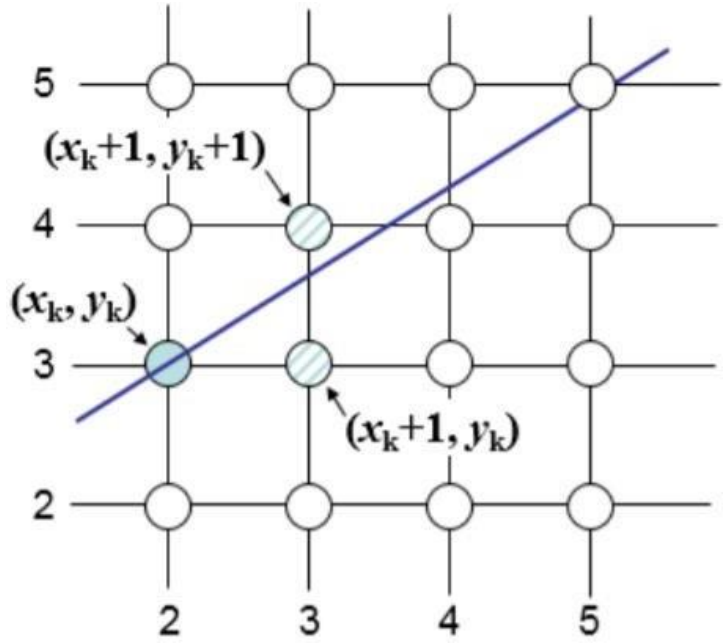
X	Y calculated	Y rounded
19	38	38
20	38.737	39
21	39.474	39
22	40.211	40
23	40.948	41
24	41.685	42
25	42.422	42
26	43.159	43
27	43.896	44
28	44.633	45
29	45.370	45
30	46.107	46
31	46.844	47
32	47.581	48
33	48.318	48
34	49.055	49
35	49.792	50
36	50.529	51
37	51.266	51
38	52.003	52

Bresenham's Line Algorithm

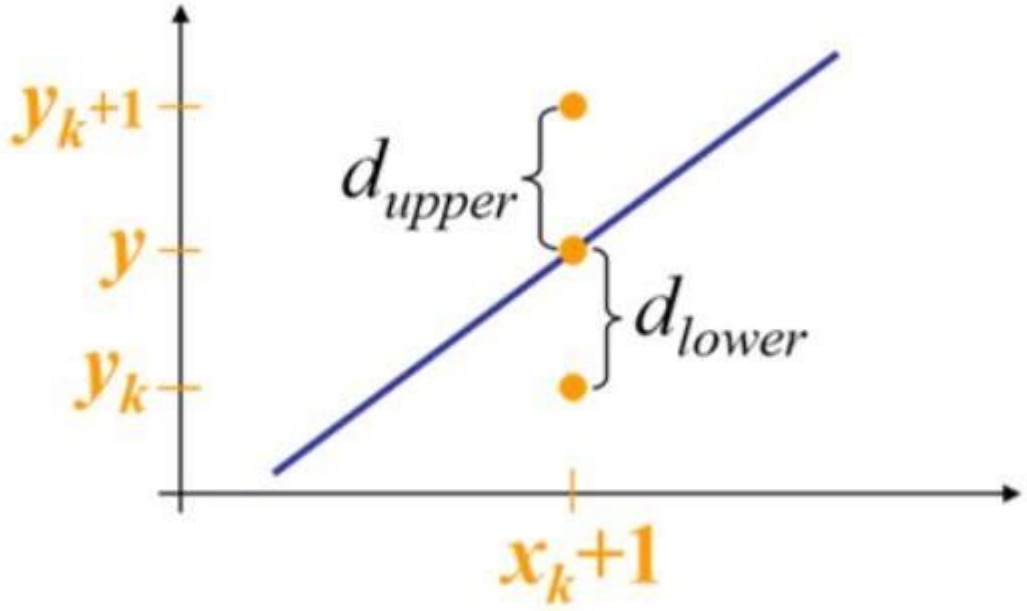
The Bresenham algorithm is another incremental scan conversion algorithm. The big advantage of this algorithm is that, it uses only integer calculations. Moving across the x axis in unit intervals and at each step choose between two different y coordinates.

An accurate and efficient raster line-generating algorithm, developed by Bresenham which scan converts line using only incremental integer calculations that can be modified to display circles and other curves.

For example, as shown in the following illustration, from position (2, 3) you need to choose between (3, 3) and (3, 4). You would like the point that is closer to the original line.



At sample position x_{k+1}, the vertical separations from the mathematical line are labelled as d_{upper} and d_{lower}.



From the above illustration, the y coordinate on the mathematical line at xk+1 is:

$Y = m(Xk + 1) + b$

So, dupper and dlower are given as follows:

$dlower = y - yk = m(Xk + 1) + b - Yk$

and

$dupper = (yk + 1) - y = Yk + 1 - m(Xk + 1) - b$

You can use these to make a simple decision about which pixel is closer to the mathematical line. This simple decision is based on the difference between the two pixel positions. d

$lower - dupper = 2m(xk + 1) - 2yk + 2b - 1$

Let us substitute m with dy/dx where dx and dy are the differences between the end-points.

$dx(dlower - dupper) = dx(2\ dy\ dx\ (xk + 1) - 2yk + 2b - 1)$

$= 2dy \cdot xk - 2dx \cdot yk + 2dy + dx(2b - 1)$

$= 2dy \cdot xk - 2dx \cdot yk + C$

So, a decision parameter pk for the kth step along a line is given by:

$P_k = dx(dlower - dupper)$

$= 2dy \cdot xk - 2dx \cdot yk + C$

The sign of the decision parameter P_k is the same as that of dlower – dupper.

If P_k is negative, then choose the lower pixel, otherwise choose the upper pixel. Remember, the coordinate changes occur along the x axis in unit steps, so you can do everything with integer calculations. At step k+1, the decision parameter is given as:

$P_k + 1 = 2dy \cdot xk+1 - 2dx \cdot yk+1 + C$

Subtracting P_k from this we get: $pk+1 - pk = 2dy(x_{k+1} - x_k) - 2dx(y_{k+1} - y_k)$

But, x_{k+1} is the same as x_k+1. So:

$P_{k+1} = P_k + 2dy - 2dx(y_{k+1} - y_k)$

Where, $y_{k+1} - y_k$ is either 0 or 1 depending on the sign of pk.

The first decision parameter p0 is evaluated at (x0, y0) is given as:

$p0 = 2dy - dx$ Now, keeping in mind all the above points and calculations, here is the Bresenham algorithm for slope m < 1:

Step 1: Input the two end-points of line, storing the left end-point in (x_0, y_0).

Step 2: Plot the point (x_0, y_0).

Step 3: Calculate the constants dx, dy, 2dy, and (2dy – 2dx) and get the first value for the decision parameter as:

$p0 = 2dy - dx$

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

If pk < 0, the next point to plot is (x_{k+1}, y_k) and $P_{k+1} = P_k + 2dy$

Otherwise, the next point to plot is (x_{k+1}, y_{k+1}) and $P_{k+1} = P_k + 2dy - 2dx$

Step 5: Repeat step 4 (dx – 1) times.

For m > 1, find out whether you need to increment x while incrementing y each time. After solving, the equation for decision parameter P_k will be very similar, just the x and y in the equation gets interchanged.

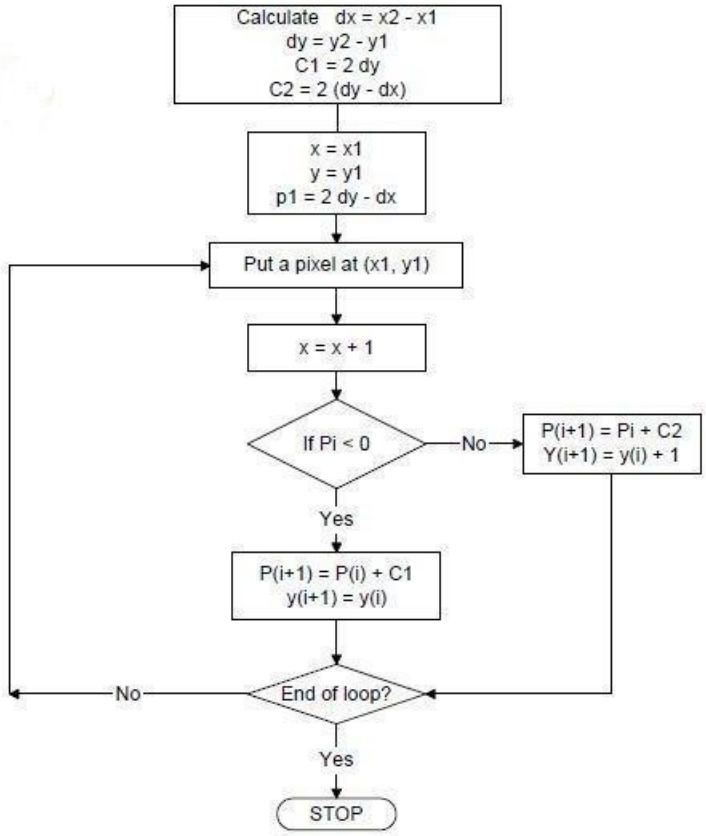


Fig. 1.28 Flow chart for line calculation procedure by Bresenham algorithm

Example 1.2

Indicate which raster locations would be chosen by Bresenham's algorithm when scan converting a line from screen coordinate (19, 38) to screen coordinate (38, 52).

$$\mathbf{m = \frac{\Delta y}{\Delta x} = \frac{y2-y1}{x2-x1} = \frac{52-38}{38-19} = 0.737}$$

So, $m \leq 1$

ΔX= 1 and $y_{i+1} = y_i + m$

Table 1.2 Calculation of pixel positions by Bresenham's Procedure

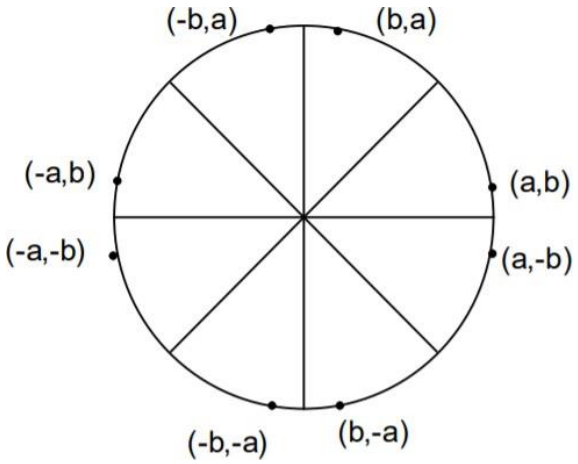
X	P	Y
19	$P_1 = 2\Delta y - \Delta x = 28 - 19 = 9$	38
20	$P_2 = P_1 + 2\Delta y - 2\Delta x = 9 + 28 - 38 = -1$	39
21	$P_3 = P_2 + 2\Delta y = -1 + 28 = 27$	39
22	$P_4 = P_3 + 2\Delta y - 2\Delta x = 27 + 28 - 38 = 17$	40
23	$P_5 = P_4 + 2\Delta y - 2\Delta x = 17 + 28 - 38 = 7$	41
24	$P_6 = P_5 + 2\Delta y - 2\Delta x = 7 + 28 - 38 = -3$	42
25	$P_7 = P_6 + 2\Delta y = -3 + 28 = 25$	42
26	$P_8 = P_7 + 2\Delta y - 2\Delta x = 25 + 28 - 38 = 15$	43
27	$P_9 = P_8 + 2\Delta y - 2\Delta x = 15 + 28 - 38 = 5$	44
28	$P_{10} = P_9 + 2\Delta y - 2\Delta x = 5 + 28 - 38 = -5$	45
29	$P_{11} = P_{10} + 2\Delta y = -5 + 28 = 23$	45
30	$P_{12} = P_{10} + 2\Delta y - 2\Delta x = 23 + 28 - 38 = 13$	46
31	$P_{13} = P_{12} + 2\Delta y - 2\Delta x = 13 + 28 - 38 = 3$	47
32	$P_{14} = P_{13} + 2\Delta y - 2\Delta x = 3 + 28 - 38 = -7$	48
33	$P_{15} = P_{14} + 2\Delta y = -7 + 28 = 21$	48
34	$P_{16} = P_{15} + 2\Delta y - 2\Delta x = 21 + 28 - 38 = 11$	49
35	$P_{17} = P_{16} + 2\Delta y - 2\Delta x = 1 + 28 - 38 = 1$	50
36	$P_{18} = P_{17} + 2\Delta y - 2\Delta x = 1 + 28 - 38 = -9$	51
37	$P_{19} = P_{18} + 2\Delta y = -9 + 28 = 19$	51
38	$P_{20} = P_{19} + 2\Delta y - 2\Delta x = 19 + 28 - 38 = 9$	52

CIRCLE GENERATION ALGORITHM

Drawing a circle on the screen is a little complex than drawing a line. There are two popular algorithms for generating a circle: Bresenham's Algorithm and Midpoint Circle Algorithm.

These algorithms are based on the idea of determining the subsequent points required to draw the circle. Let us discuss the algorithms in detail:

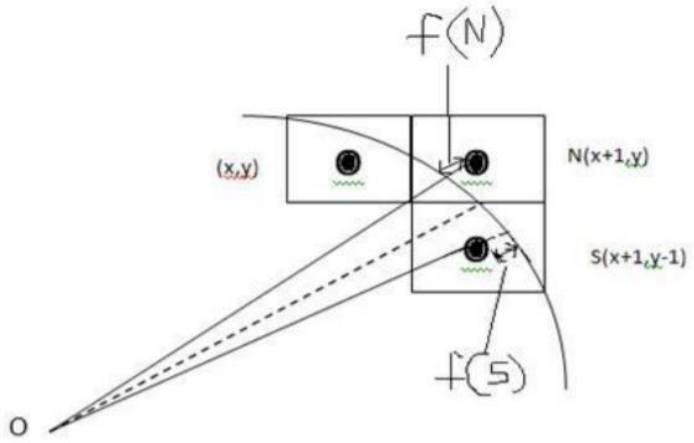
The equation of circle is $X^2 + Y^2 = r^2$, where r is radius.



Bresenham's Algorithm

We cannot display a continuous arc on the raster display. Instead, we have to choose the nearest pixel position to complete the arc.

From the following illustration, you can see that we have put the pixel at (X, Y) location and now need to decide where to put the next pixel: at N (X+1, Y) or at S (X+1, Y-1).



This can be decided by the decision parameter d.

- If d <= 0, then N(X+1, Y) is to be chosen as next pixel.
- If d > 0, then S(X+1, Y-1) is to be chosen as the next pixel.

Algorithm

Step 1: Get the coordinates of the center of the circle and radius, and store them in x, y, and R respectively. Set P=0 and Q=R.

Step 2: Set decision parameter D = 3 – 2R.

Step 3: Repeat through step-8 while X < Y.

Step 4: Call Draw Circle (X, Y, P, Q).

Step 5: Increment the value of P.

Step 6: If D < 0 then D = D + 4x + 6.

Step 7: Else Set Y = Y + 1, D = D + 4(X-Y) + 10.

Step 8: Call Draw Circle (X, Y, P, Q).

```
Draw Circle Method(X, Y, P, Q).
Call Putpixel (X + P, Y + Q).
Call Putpixel (X - P, Y + Q).
Call Putpixel (X + P, Y - Q).
Call Putpixel (X - P, Y - Q).
Call Putpixel (X + Q, Y + X).
Call Putpixel (X - Q, Y + X).
Call Putpixel (X + Q, Y - X).
Call Putpixel (X - Q, Y - X).
```

GRAPHICS EXCHANGE STANDARDS

The need for portability of the geometric model among different hardware platforms has led to the development of device independent graphics.
Simultaneously standards for the exchange of drawing database among software packages have been evolved to facilitate the integration of design and manufacturing operations.

Introduction

The heart of any CAD model is the component database. This includes the graphics entities like points, lines, arcs, circles etc. and the co-ordinate points, which definethe location of these entities.

This geometric data is used in all downstream applications of CAD, which include finite element modeling and analysis, process planning, estimation, CNC programming, robot programming, programming of co-ordinate measuring

machines, ERP system programming and simulation.

In order to achieve at least a reasonably high level of integration between CAD, analysis and manufacturing operations, the component database must contain:

i. Shapes of the components (based on solid models)
ii. Bill of materials (BOM), of the assembly in which the components are used.
iii. Material of the components
iv. Manufacturing, test and assembly procedures to be carried out to produce a component so that it is capable of functioning as per the requirements of design.

In designing a data structure for CAD database the following factors are to be considered:

i. The data must be neutral
ii. The data structure must be user-friendly
iii. The data must be portable.

In order to achieve the above requirements, some type of standardization has to be followed by the CAD software designers. The basic elements associated with a CAD system are:

• Operator (user)

• Graphics support system

• Other user interface support system

• Application functions

• Database

A diagrammatic presentation of these elements is given in Figure 1.25

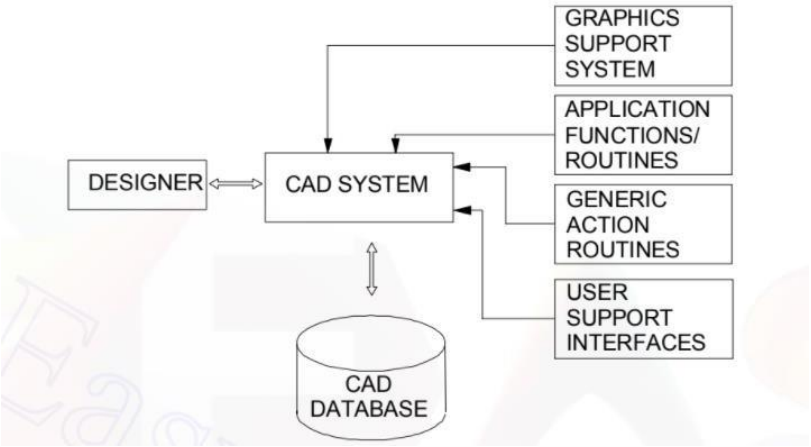


Fig. 1.25 CAD System

The reasons for evolving a graphic standard thus include:

• Need for exchanging graphic data between different computer systems.

• Need for a clear distinction between modeling and reviewing aspects.

So proper interface standards are required to be established, some of them are

• GKS (Graphics Kernal System)

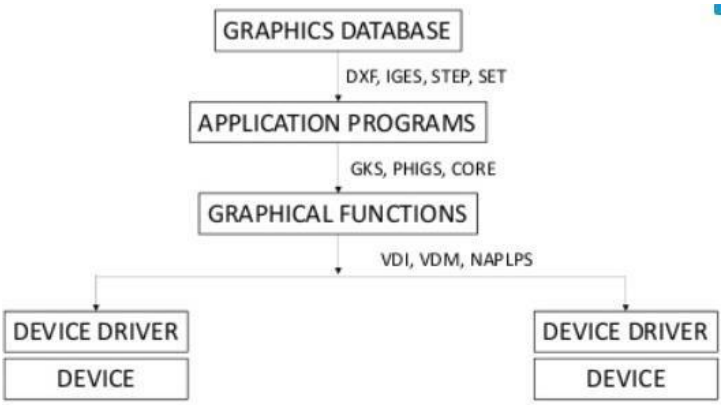
• PHIGS (Programmer's Hierarchical Interface for Graphics)

• GKS-3D

• IGES

• STEP

• DXF



Standards for graphics programming

Attempts to develop a graphics standard resulted in the following developments in 70's:

i. A Graphic Standards Planning Committee (GSPC) was formed in 1974 by ACM-SIGGRAPH (Association of Computing Machinery's Special Interest Group on Graphics and Interactive Techniques).

ii. A committee for the development of computer graphics standard was formed by DIN in 1975.

iii. IFIP organized a workshop on Methodology in Computer Graphics in 1976. iv. A significant development in CAD standards is the publication of Graphical Kernel System (GKS) in 1982.

1. GKS (Graphics kernel system): It is an ANSI (American National Standards Institute and ISO (International Standards Organization) standard. It interfacesthe application program with graphics support package.

2. IGES (Initial graphics exchange specification): It is an ANSI standard. It enables an exchange of model database among CAD/CAM software.

3. PHIGS (Programmer's hierarchical interactive graphics system): It supports workstations and their related CAD/CAM applications. It supports 3-dimensional modeling of geometry segmentation and dynamic display.

4. CGM (Computer graphics metafile): It defines functions needed to describe an image. Such description can be stored or transported from one graphics device to another.

S. CGII (Computer graphics interface): It is designed to interface plotters to GKSor PHIGS. It is the lowest device independent interface in a graphics system.

6. Drawing Exchange Format (DXF): The DXF format has been developed and supported by Autodesk for use with the AutoCAD drawing files. It is not an industry standard developed by any standards organisation, but in view of the widespread use of AutoCAD made it a default standard for use of a variety of CAD/CAM vendors. A Drawing Interchange File is simply an ASCII text file with a file extension of .DXF and specially formatted text.

7. Standard for the Exchange of Product Model Data (STEP), officially the ISO standard 10303, Product Data Representation and Exchange, is a series of International Standards with the goal of defining data across the full engineeringand manufacturing life cycle. The ability to share data across applications, across vendor platforms and between contractors, suppliers and customers, is the main goal of this standard.

8. Parasolid: It is a portable "kernel" that can be used in multiple systems - bothhigh-end and mid-range. By adopting Parasolid, start-up software companies have eliminated a major barrier to application development - a high initial investment. This enabled them to effectively market softwares with strong solidmodeling functionality at lower-cost.

9. PDES (Product Data Exchange Specification) is an exchange for product data Insupport of Industrial automation. "Product data" encompasses data relevant to the entire life cycle of a product such as design, manufacturing, quality assurance, testing and support. In order to support industrial automation, PDES files are fully Interpretable by computer. For example, tolerance Information would be carried In a form directly interpretable by a computer rather than a computerized text form which requires human intervention to interpret.

Features of GKS

GKS (Graphical Kernel System) is an ANSI and ISO standard. GKS standardizes two-dimensional graphics functionality at a relatively low level. The primary purposes of the standard are:

• To provide for portability of graphics application programs.

• To aid in the understanding of graphics method by application programmers.

• To provide guidelines for manufacturers in describing useful graphicscapabilities.

The GKS (ANSI X3.124-1985) consists of three basic parts:

i. An informal exposition of contents of the standard, which includes such things as positioning of text, filling of polygons etc.

ii. A formalization of the expository material outlined in (i) by way of abstracting the ideas into functional descriptions (input/output parameters, effect ofeach function etc.).

iii. Language bindings, which are the implementations of the abstract functions, described in (ii) in a specific computer language like FORTRAN,Ada or C.

Figure 1.26 shows the GKS implementation in a CAD workstation.

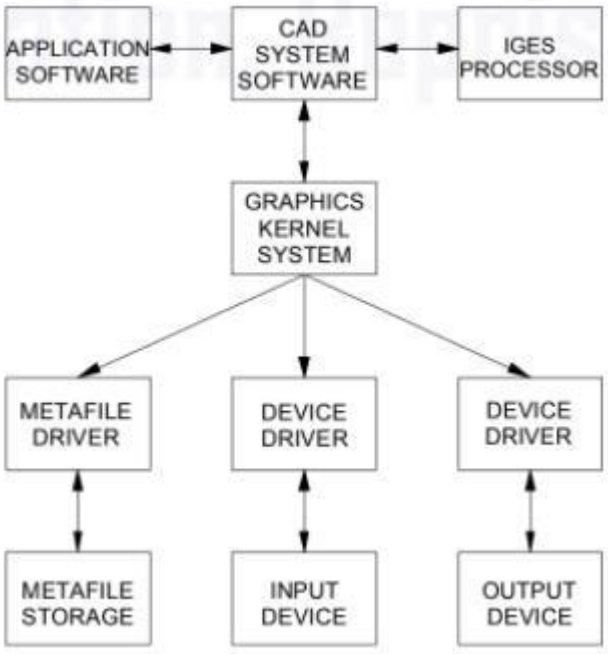


Fig. 1.26: GKS implementation in a CAD System

The features of GKS include:

i. Device independence: The standard does not assume that the input or output devices have any particular features or restrictions.
ii. Text/Annotations: All text or annotations are in a natural language likeEnglish.
iii. Display management: A complete suite of display managementfunctions, cursor control and other features are provided.
iv. Graphics Functions: Graphics functions are defined in 2D or 3D.

GKS offers two routines to define the user created pictures. They are primitive functions and attribute functions.

Examples of primitive functions are: • POLYLINE to draw a set of connected straight-line vectors

• POLYMARKER to draw a set of markers or shapes

• FILL AREA to draw a closed polygon with interior fill

• TEXT to create characters

• GDP (Generalized Drawing Primitive) to specify the standard drawing entities likecircle, ellipse etc.

The attribute functions define the appearance of the image e.g. color, line-type etc.Current level of GKS is GKS-3D, which provides several other functions. GKS-3D is anextension to GKS, which allows the production of 3-D objects.

Exchange of CAD data between software packages

Necessity to translate drawings created in one drafting package to another often arises. For example you may have a CAD model created in PRO/E package and youmay wish that this might be transferred to I-DEAS or Unigraphics.

It may also be necessary to transfer geometric data from one software to another. This situation arises when you would want to carry out modeling in one software, say PRO/E and analysis in another software, say ANSYS.

One method to meet this need is to write direct translators from one software to another. This means that each system developer will have to produce its own translators. This will necessitate a large number of translators. If we have three software packages we may require six translators among them.

This is shown in Fig. 1.27.

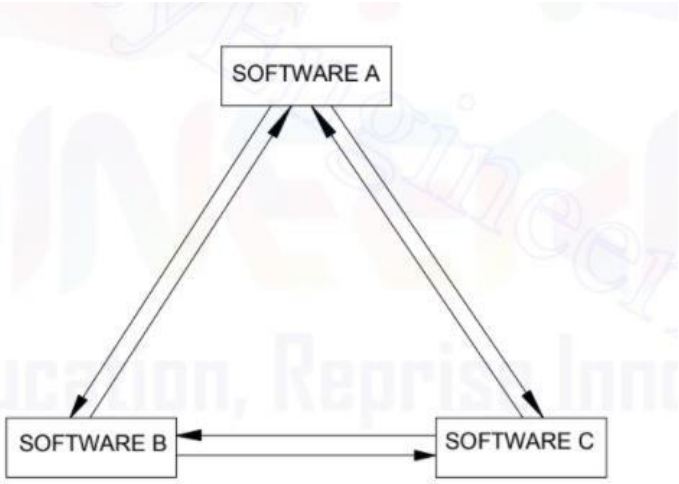


Fig. 1.27: Direct Data Translation

A solution to this problem of direct translators is to use neutral files. These neutral files will have standard formats and software packages can have pre-processors to convert drawing data to neutral file and postprocessors to convert neutral file data to drawing file.

Figure 1.28 illustrates how the CAD data transfer is a accomplished using neutral file.

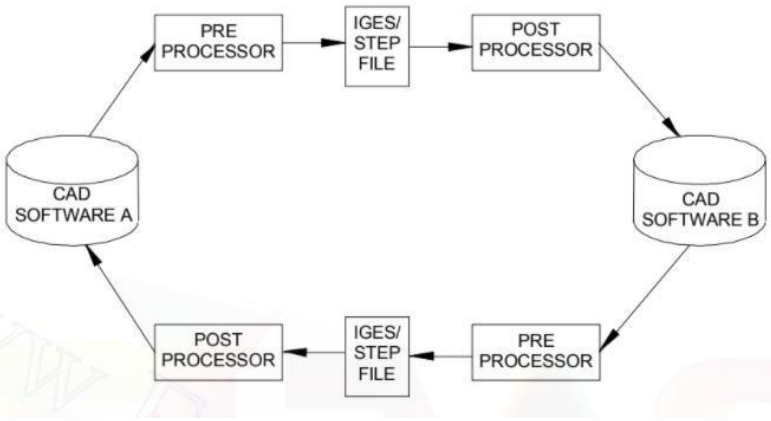


Fig. 1.28: CAD Data Exchange using Neutral files

Three types of neutral files are:

i. Drawing exchange files (DXF)

ii. IGES files

iii. STEP files

Drawing exchange files (DXF)

DXF is short for Drawing Interchange Format or Drawing Exchange Format. Commonly known as AutoCAD DXF format, is a CAD data file format. It was developed by Autodesk to enable data interoperability between AutoCAD and other programs. DXF file format was first introduced in 1982. The key purpose of this format was to produce the exact representation of AutoCAD native DWG files on other applications. For many years, the importing process of DXF files had been very difficult because of the unavailability of specifications of the format. Presently the format specifications are available as PDF but this format is rarely used in AutoCAD applications.

Initial Graphics Exchange Specification (IGES) Graphics Standard

The IGES committee was established in the year 1979. The CAD/CAM Integrated Information Network (CIIN) of Boeing served as the preliminary basis of IGES. IGES version 1.0 was released in 1980. IGES continues to undergo revisions. IGES is a popular data exchange standard today.

IGES files can also be generated for:

i. Surfaces

ii. Datum curves and points

Standard for the Exchange of Product Data (STEP)

Seamless exchange of product data is critical to CAD/CAM/CAE systems. The Standard for the Exchange of Product Data (STEP) is the enabler for such seamless data exchange.

It provides a worldwide standard for storing, sharing and exchanging product information among different CAD systems. Although STEP itself is the basis for Product Data Management System (PDM). It covers border functionalities. It includes methods of representing all critical product specifications such as shape information, materials, tolerances, finishes and product structure.

Whereas the Initial Graphics Exchange Specification (IGES) standard has widespread use, it has its shortcomings. It does not convey the extensive product information needed in the design and manufacturing cycle. Often IGES translators are required to move design data from one CAD system to another. STEP is often viewed as a replacement for IGES, though IGES is still expected to be in active use for some more time in the future. Although the current focus of STEP is on mechanical parts, STEP is a data exchange standard that would apply to a wide range of product areas, including electronics, architectural, engineering and construction, apparel and shipbuilding.

DATABASE MANGEMENT SYSTEM

The data generated during a design and engineering process shares multiple features and structures. The more complex the product and the longer the project the more important is the control of the product configuration data. Apart from the traditional drawings and CAD models this data includes various technical notes, measurement results, material properties, key parameters of the system, manufacturing and assembly instructions, organizational relations (i.e. responsibilities) and chronological change history of each component of the product.

The role of Engineering Data Management (EDM) is to store and provide a user- friendly access to this data during the product's life-cycle according to certain predefined control rules. These rules are defined by the configuration management team of the LHC-project in collaboration with the key players of the project.

Engineering Data Management Systems

In recent years, there has been a growing awareness that although computer-aideddesign technology and the widespread use of computer applications to prepare engineering documents were accomplishing their objective of improving the productivity of individual engineers and designers, they were not doing much to improve the productivity of the overall enterprise.

To accomplish this latter objective better methods were needed to share information between members of the design teams and other groups involved inthe product life cycle.

Engineering Data Management (EDM) is a set of techniques and tools to organize, control and distribute all product related data during a product's life cycle (Fig.
1.29). They have been developed to reduce the development cycle of new productswhile maintaining control of the data and distributing it automatically to the peoplewho need it when they need it.

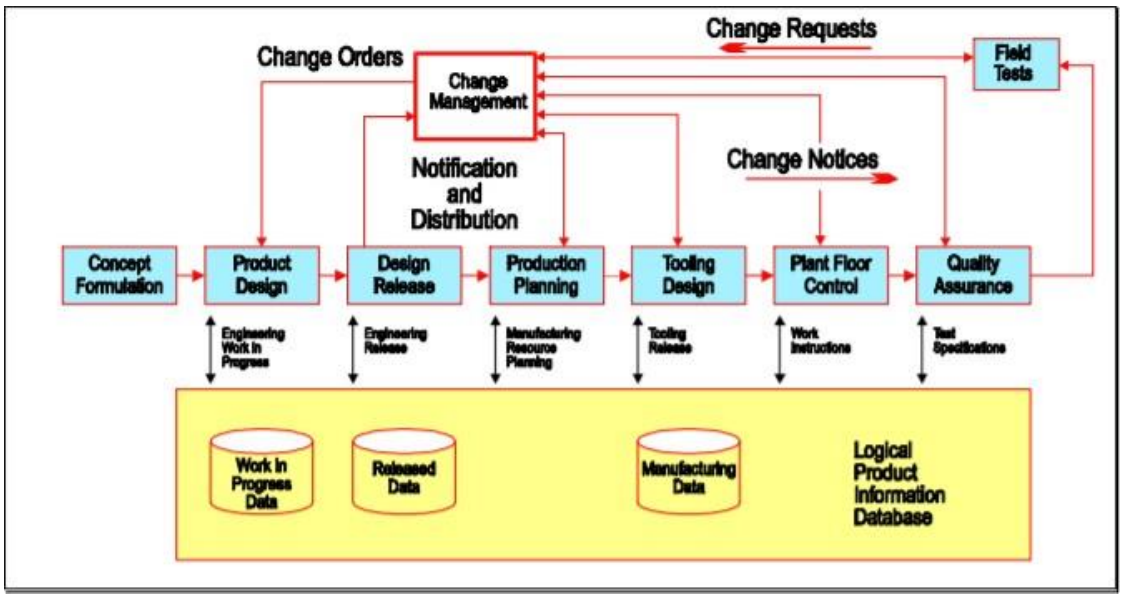


Fig 1.29: EDM's role in the Product Life Cycle

One of the prime motivation for EDM is the sheer volume and diversity of engineering data.

An exhaustive description of a part or an assembly requires a huge amount of documentation. The more complex the item, the more diverse is the documentation. The following information topics could easily be listed under the heading of the complete set of the engineering documentation of a part or an assembly:

- drawings,
- CAD models,
- part lists,
- structural analysis models,
- spread sheets,
- technical notes,
- measurements results,
- manufacturing instructions,
- assembly instructions,
- organizational relations,
- material properties,
- schematics,
- applicable standards,
- photos and shaded images

The basic concept of EDM to manage datasets created by the various tools (Fig. 1.30) is to create a metadata layer, i.e. a layer containing pointers and summary information for the datasets being managed. The metadata itself is usually maintained in a database and managed by a DBMS. This is similar to the use of library card catalogs.

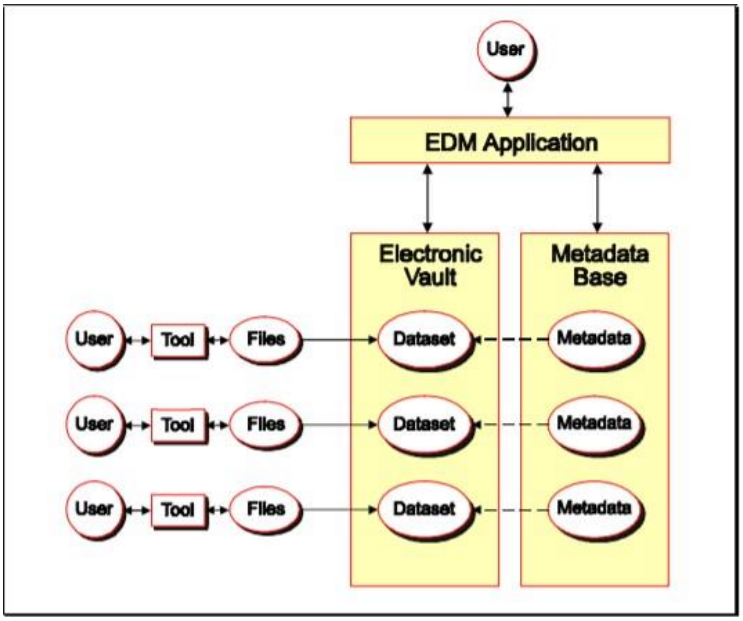


Fig 1.30: EDM Basic Concept

Tutorial Questions

1. Explain the applications where computers are used in Industrial Manufacturing.
2. Explain about Product Cycle.
3. Explain basic structure of CAD/CAM Hardware.
4. Explain about different display devices.
5. Explain fundamentals of LCD and LED.
6. What are the different types of coordinate systems? Explain them.
7. Write DDA and Bresenham's Algorithm for Line generation.
8. Write DDA and Bresenham's Algorithm for Curve generation.
9. Explain about different Graphic Exchange Standards.
10. Explain about Database Management System.

QUESTION BANK

1. Explain about Product Cycle.
2. Explain about different display devices.
3. Explain fundamentals of LCD and LED.
4. Write DDA and Bresenham's Algorithm for Line and Curve generation.
5. Explain about different Graphic Exchange Standards.

UNIT 2

CURVES AND SURFACES

Course Objective:

Understand the Mathematical representations of curves and surfaces used in geometric construction.

Course Outcome:

Understand and develop the Mathematical representations of curves used in geometric construction.

NO OF LECTURE HOURS: 9

LECTURE	LECTURE TOPIC	KEY ELEMENTS	LEARNING OBJECTIVES (2 to 3 objectives)
1.	Curves Representation	Representation of curves	Understanding how to represent the curves (B5, B6)
2.	Parametric representation of lines: Locating a point on a line, parallel lines,	Parametric representation of lines	Learning about representation of lines (B4, B5)
3.	Perpendicular lines, distance of a point, Intersection of lines, Parametric representation of circle	Parametric representation of lines & circle	Learning about representation of lines & circle (B4, B5)
4.	Parametric representation of ellipse, parabola, hyperbola	Parametric representation of ellipse, parabola, hyperbola	Learning about representation of ellipse, parabola, hyperbola (B4, B5)
5.	Synthetic Curves: Concept of continuity,	Synthetic curves	Learning about Synthetic Curves: Concept of continuity (B4, B5)
6.	Bezier Curve: equations, properties	Bezier Curves	Learning about Bezier Curve: equations, properties (B4, B5)
7.	Properties and advantages of B-Splines and NURBS	B-Splines and NURBS	Learning about Properties and advantages of B-Splines and NURBS (B5, B6)

8.	Various types of surfaces along with their typical applications	Representation of surfaces	Learning about Various types of surfaces along with their typical applications (B4, B5)
9.	Various types of surfaces along with their typical applications	Algorithms for curve generation	Learning about Various types of surfaces along with their typical applications (B4, B5)

Curve Representation

- Curve is defined as the locus of point moving with one degree freedom.
- A curve can be represented by following two methods either by storing its analytical equation or by storing an array of co-ordinates of various points
- Curves can be described mathematically by following methods:
  - (i) Non-parametric form
    - a) Explicit form
    - b) Implicit form
  - (ii) Parametric form

Non-parametric form:

- In this, the object is described by its co-ordinates with respect to current reference frame in use.

Explicit form: (Clearly expressed)

- In this, the co-ordinates of y and z of a point on curve are expressed as two separate functions of x as independent variable.

$$P = [x \quad y \quad z]$$
$$= [x \quad f(x) \quad g(x)]$$

Implicit form: (Not clearly expressed)

- In this, the co-ordinates of x, y and z of a point on curve are related together by two functions.

$$F(x, y, z) = 0$$
$$G(x, y, z) = 0$$

Limitation of nonparametric representation of curves are:

1. If the slope of a curve at a point is vertical or near vertical, its value becomes infinity or very large, a difficult condition to deal with both computationally and programming-wise. Other ill-defined mathematical conditions may result.
2. Shapes of most engineering objects are intrinsically independent of any coordinate system. What determines the shape of an object is the relationship between its data points themselves and not between these points and some arbitrary coordinate system.
3. If the curve is to be displayed as a series of points or straight line segments, the computations involved could be extensive.

Parametric form:

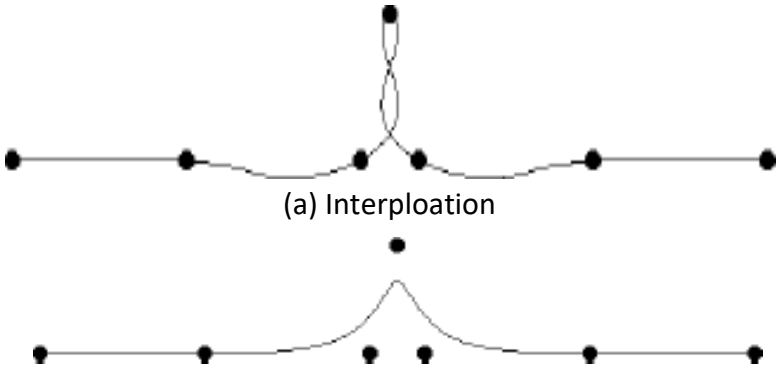
- In this, a parameter is introduced and the co-ordinates of x, y and z are expressed as functions of this parameters. This parameter acts as a local co-ordinate for points on curve.

$$P(u) = [x \quad y \quad z]$$
$$= [x(u) \; y(u) \; z(u)]$$

Classification of Curves

- Curves can be classified as shown below mainly in two categories:

1. **Analytical Curves:**
   - The curves which are defined by the analytical equations are known as analytical curves.
   - Examples: line, circle, ellipse, parabola, hyperbola etc.
2. **Synthetic Curves:**
   - The curves which are defined by set of data points are known as synthetic curves.
   - Combination of polynomial segments to represent a desired curve is called a synthetic curve.
   - These curves can be conveniently twisted, shaped or bent by changing the control points.
   - Examples: Cubic spline curve, B-spline curve, Bezier curve etc.
   - Synthetic curves take up where the analytic curves leave – the latter are not that efficient at geometric design of mechanical parts.
   - Need for synthetic curves arise in two occasions:
     - When a curve is represented by a collection of measured datapoints,
     - When an existing curve must change to meet new design requirements the designer needs a curve representation that is directly related to the data points and is flexible enough to bend, twist or change the shape by changing one or more data points.
   - Some examples of complex geometric design are: Car bodies, Ship hulls, Airplane fuselage and wings, Propeller blades, Shoe insoles, aesthetically designed bottles etc.
   - When the curve pass through all the data points, then the curve is known as Interpolant curve. It sounds more logical but can be more unstable and ringing.
   - When a smooth curve is approximated through the data points, then the curve is known as approximation curve. It turns out to be convenient.



(a) Interploation

(b) Approximation

Fig. 2.1 Interpolant and Approximated Curves

- Synthetic curve pass through defined data points and can thus represented by polynomial equations.
- The parametric form for any synthetic curve is:

  $P(u) = C_3 u^3 + C_2 u^2 + C_1 u + C_0$

  where, u = Parameter and C_i = Polynomial coefficients

Parametric representation of analytical curve

- **Lines**

- A line can be represented in a parametric form by an input of two endpoints or by one point and a length and direction. Besides these, modifiers and filters can be used to represent these lines. The two important methods of establishing equations for lines are considered.

- **A line connecting two endpoints in parametric form, having the values of parameter *u* as 0 and 1 at the endpoints:**

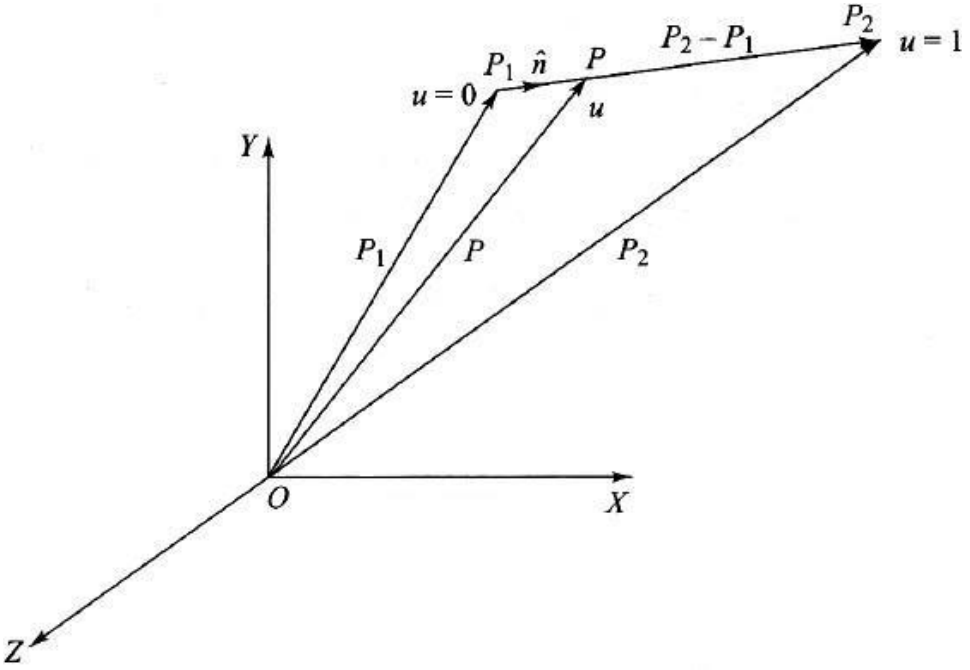


Fig. 2.2 Line between Two Endpoints

- Consider a line between two given endpoints P_1 (x_1, y_1, z_1) and P_2(x_2, y_2, z_2) as shown in Fig. 2.2. The position vectors for these two points would be P_1 and P_2. Consider any point P (x, y, z) on the line P_1 P_2.
- Let a parametric system be introduced such that the point P is represented by the parameter *u* such that its values are 0 and 1 at endpoints P_1 and P_2 respectively.
- From ΔOPP_1, we can express the vector between endpoints P_1 and P as P - P_1. This vector represents the parameter *u*. The mapping between the Cartesian and parametric space can be done by the following relation :

$$\frac{\text{Distance of P from } P_1 \text{ in cartesian space}}{\text{Distance of P from } P_1 \text{ in parametric space}} = \frac{\text{Endpoint Length in cartesian space}}{\text{Endpoint Length in parametricspace}}$$

$$\frac{P-P_1}{u}=\frac{P_2-P_1}{1}$$

$$P-P_1=u(P_2-P_1)$$

$$P=P_1+u(P_2-P_1) \qquad \text{where, } 0 \le u \le 1$$

- This equation represents the vector equation of the line in parametric form. In scalar form, it can be written as:

$$x=x_1+u(x_2-x_1)$$

$$y=y_1+u(y_2-y_1)$$

$$z=z_1+u(z_2-z_1) \qquad \text{where, } 0 \le u \le 1$$

- Any point on the above line or its extension can be found by substituting the value of parameter *u*. The tangent vector of the line would be represented by :

$$P' = P_2 - P_1$$

- In scalar form these would have the coordinates expressed by the following equations:

$$x' = x_2 - x_1$$

$$y' = y_2 - y_1$$

$$z' = z_2 - z_1$$

- It can be seen that the tangent vector is independent of the parameter *u*, thus representing the constant slope of the line. Above equations are sufficient to represent all properties of the line for CAD/CAM representation. The length L and direction vector $\hat{n}$ can be calculated using the following equations :

$$L = |P_2 - P_1|$$

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\hat{n} = \frac{P_2 - P_1}{L}$$

- **A line passing through a point and defined by a length and direction:**

- Consider a case, where a line is to pass through a point P_1 and has its length L and direction vector $\hat{n}$ defined as input as shown in Fig. 2.3. The computation of the endpoint of this line needs to be done based on the above inputs. As seen earlier,

$$\hat{n} = \frac{P - P_1}{L}$$

$$P = P_1 + L\left(\hat{n}\right)$$

where, $-\infty \le L \le \infty$

- Once the user inputs P_1, L and $\hat{n}$, the endpoint can be calculated as per the above equation. The two endpoints can then be expressed in parametric form with values of parameter *u* as 0 and 1.

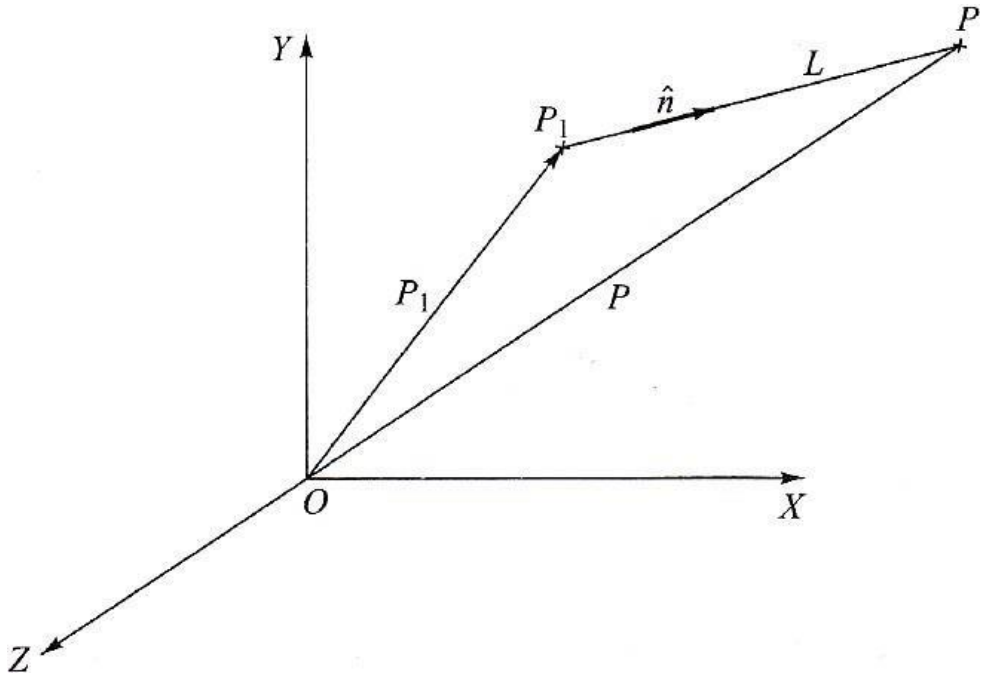


Fig. 2.3 Line passing through a point and defined by a Length and Direction

- **Parallel Lines**

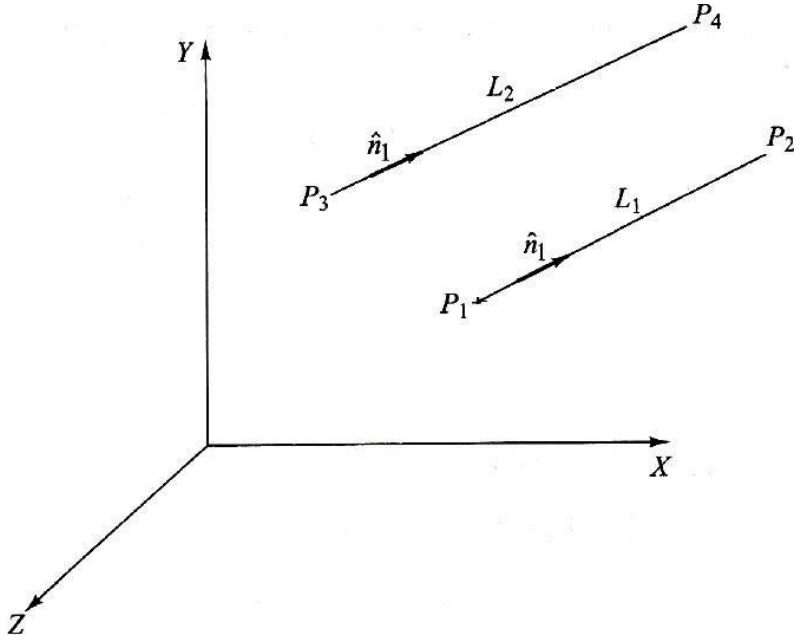


Fig. 2.4

- Assume that the existing line has the two endpoints P_1 and P_2, a length L_1, and a direction defined by the unit vector $\hat{n}_1$. The new line has the same direction $\hat{n}_1$, a length L_2 and endpoints P_3 and P_4.
- The unit vector $\hat{n}_1$ is givenby

$$\hat{n} = \frac{P_2 - P_1}{L_1}$$

- The unit vector for the parallel line will be same as unit vector for the existingline.

$$\hat{n} = \frac{P_4 - P_3}{L_2}$$

$$P_4 = P_3 + L_2 \hat{n}$$

- The equation of parallel line becomes,

$$P = P_3 + u(P_4 - P_3)$$

- **Angle between two lines**

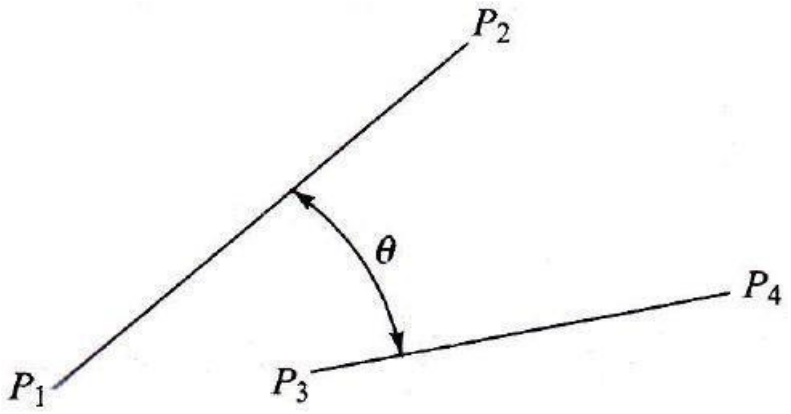


Fig. 2.5

- Fig. 2.5 represents two lines L_1 and L_2 having endpoints are P_1, P_2 and P_3, P_4, the angle measurement command uses the equation

$$\cos\theta = \frac{(P_2 - P_1)\cdot(P_4 - P_3)}{|P_2 - P_1|\,|P_4 - P_3|}$$

$\cos\theta = L_1 \,.\, L_2$

- **Distance of a point from line**

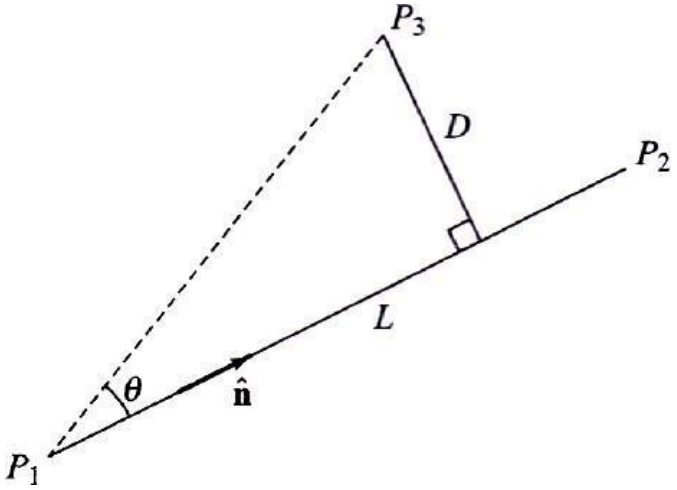


Fig. 2.6

- Fig. 2.6 shows the distance D from point P_3 to the line whose endpoints are P_1 and P_2 and direction is $\hat{n}$. Its length is L. D is given by

$$D = \left|(P_3 - P_1) \times \hat{n}\right| = \left|(P_3 - P_1) \times \frac{(P_3 - P_1)}{|P_3 - P_1|}\right|$$

Example 2.1: For the position vectors $P_1[1\ 2]$ and $P_2[4\ 3]$, determine the parametric representation of line segment between them. Also determine the slope and tangent vector of line segment.

A parametric representation of line is

$$P = P_1 + u(P_2 - P_1)$$

$$= [1\ 2] + u([4\ 3] - [1\ 2])$$

$$= [1\ 2] + u[3\ 1]$$

Parametric representation of x and y components are

$$x(u) = x_1 + u(x_2 - x_1)$$

$$= 1 + 3u$$

$$y(u) = y_1 + u(y_2 - y_1)$$

$$= 2 + u$$

The tangent vector is obtained by differentiating P(u)

$$P'(u) = *x'(u)\ y'(u)+$$

$$= [3\ 1]$$

The slope of line segment is

$$\frac{dy}{dx} = \frac{dy/du}{dx/du}$$

$$= \frac{y'}{x'}$$

$$= \frac{1}{3}$$

Example 2.2 : The end points of line are P1(1,3,7) and P2(–4,5, –3). Determine

i. Tangent vector of the line
ii. Length of line
iii. Unit vector in the direction of line

Parametric representation of x, y and z components are

$$x(u) = x_1 + u(x_2 - x_1)$$

$$= 1 - 5u$$

$$y(u) = y_1 + u(y_2 - y_1)$$

$$= 3 + 2u$$

$$z(u) = z_1 + u(z_2 - z_1)$$

$$= 7 - 10u$$

Tangent vector, $P'(u) = *x'(u)\ y'(u)\ z'(u)+$

$$= [-5\ 2\ -10]$$

$$= -5i + 2j - 10k$$

Length of line, $L = |P_2 - P_1|$

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$= \sqrt{(-4-1)^2 + (5-3)^2 + (-3-7)^2}$$

= 11.358

Unit vector in the direction of line, $\hat{n} = \frac{P2-P1}{|P2-P1|} = \frac{P2-P1}{L} = \frac{1}{11.358}$ [-5 2 -10]

= [-0.44 0.176 -0.88]

= -0.44i + 0.176j – 0.88k

Example 2.3: A line is represented by the end point P_1(2, 4, 6) and P_2(-3, 6, 9). If the value of parameter u at P_1 and P_2 is 0 and 1 respectively, determine the tangent vector for the line. Also determine the coordinate of a point represented by; u equal to 0, 0.25, -0.25, 1 and 1.5. Also find the length and unit vector of line between two points P_1 and P_2.

Parametric representation of x, y and z components are

$x(u) = x_1 + u(x_2 - x_1)$

$= 2 - 5u$

$y(u) = y_1 + u(y_2 - y_1)$

$= 4 + 2u$

$z(u) = z_1 + u(z_2 - z_1)$

$= 6 - 3u$

Tangent vector, P'(u) = *x'(u) y'(u) z'(u)+

= [–5 2 –3]

= –5i + 2j –3k

u	0	0.25	– 0.25	1	1.5
x (u)	2	0.75	3.25	–3	–5.5
y (u)	4	4.5	3.5	6	7
z (u)	6	5.25	6.75	3	1.5

Length of line, $L = |P_2 - P_1|$

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$= \sqrt{(-3-2)^2 + (6-4)^2 + (9-6)^2}$$

= 6.16

Unit vector in the direction of line, $\hat{n} = \frac{P2-P1}{|P2-P1|} = \frac{P2-P1}{L} =$

$\frac{1}{6.16}$ [-5 2 -3] = [-0.81 0.324 -0.487]

= -0.8i + 0.324j – 0.487k

➢ **Circles**

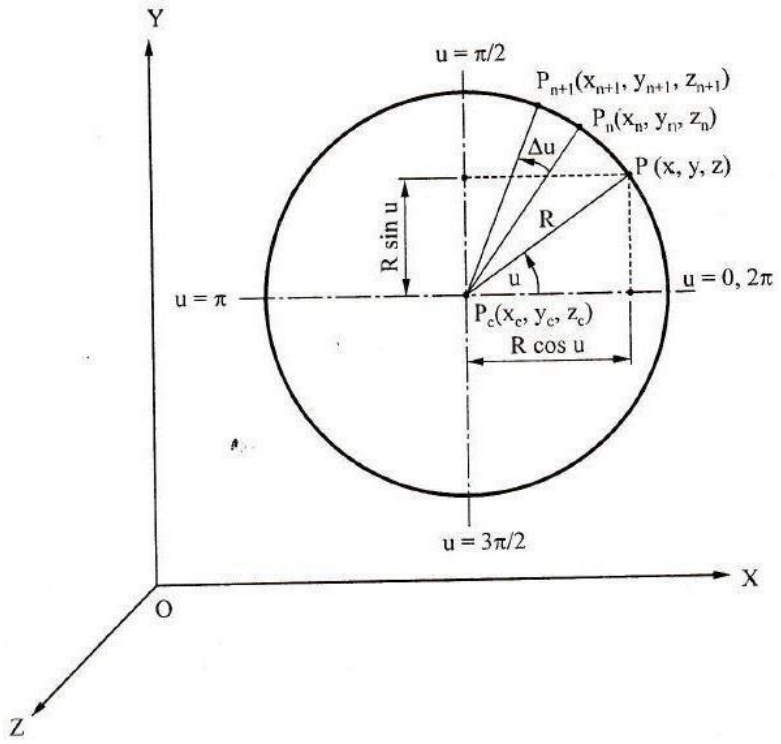


Fig. 2.7 Circle in Parametric Form

- A circle is represented in the CAD/CAM database by storing the values of its center and its radius. In a parametric form, a circle shown in Fig. 2.7, can be represented by the following set of equations:

$$x = x_c + R \ \text{Cos}\ u$$

$$y = y_c + R \ \text{Sin}\ u$$

$$z = z_c$$

where, $0 \le u \le 2\pi$

- In this case the parameter *u* represents the angle measured from the X axis to any point P on the circumference of the circle. By incrementing the values of *u* from 0 to 2л, the intermediate points on the circle can be worked out.
- These points can then be connected by straight lines for the purpose of display. However, this method of computation is inefficient as it requires trigonometric evaluation of functions.
- A convenient method to determine the different points on the circle is to use the incremental equations. Consider any point P_n (x_n, y_n, z_n) on the circle.
- Let the subsequent incremental point be P_{n+1} (x_{n+1}, y_{n+1}, z_{n+1}). From the above equation, we get:

$$x_n = x_c + R \ \text{Cos}\ u$$

$$\therefore R \ \text{Cos}\ u = x_n - x_c \qquad (2.1)$$

$$y_n = y_c + R \ \text{Sin}\ u$$

$$\therefore R \ \text{Sin}\ u = y_n - y_c \qquad (2.2)$$

Similarly, $x_{n+1} = x_c + R\ Cos\ (u + \Delta u)$ (2.3)

$y_{n+1} = y_c + R\ Sin\ (u + \Delta u)$ (2.4)

$z_{n+1} = z_n$ (2.5)

From (2.3), (2.1) & (2.2)

$x_{n+1} = x_c + R\ Cos\ u Cos\ \Delta u - R\ Sin\ u Sin\ \Delta u$

$\therefore x_{n+1} = x_c + (x_n - x_c) Cos\ \Delta u - (y_n - y_c) Sin\ \Delta u$ (2.6)

From (2.4), (2.1) & (2.2)

$y_{n+1} = y_c + R\ Sin\ u Cos\ \Delta u + R\ Cos\ u Sin\ \Delta u$

$\therefore y_{n+1} = y_c + (y_n - y_c) Cos\ \Delta u + (x_n - x_c) Sin\ \Delta u$ (2.7)

And $z_{n+1} = z_c$

- Thus, the circle can start from an arbitrary point and successive points with equal spacing can be calculated. Cos Δu & Sin Δu have to be calculated only once, this eliminates the computations of trigonometric functions for each point.
- This algorithm is useful for hardware implementation to speed up the circle generation & display.
- If two endpoints of diameter of circle are given, then the center & radius of circle are calculated as below.
- Let $P_1 (x_1, y_1, z_1)$ & $P_2 (x_2, y_2, z_2)$ and center point $P_c (x_c, y_c, z_c)$

$$\text{Radius, } R = \frac{1}{2}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\text{Center, } P_c = \frac{1}{2}(P_1 + P_2)$$

$$[x_c \quad y_c \quad z_c] = \left[\frac{x_1+x_2}{2} \quad \frac{y_1+y_2}{2} \quad \frac{z_1+z_2}{2}\right]$$

Example 2.4: The two endpoints of diameter of a circle are $P_1(13,15,7)$ and $P_2(35,40,7)$. Determine the centre and radius of circle.

The centre of a circle,

$$P_c = \frac{1}{2}(P_1 + P_2)$$

$$[x_c \quad y_c \quad z_c] = \left[\frac{x_1+x_2}{2} \quad \frac{y_1+y_2}{2} \quad \frac{z_1+z_2}{2}\right]$$

$$[x_c \quad y_c \quad z_c] = \left[\frac{13+35}{2} \quad \frac{15+40}{2} \quad \frac{7+7}{2}\right]$$

$$[x_c \quad y_c \quad z_c] = [24 \quad 27.5 \quad 7]$$

The radius of circle $R = \frac{1}{2}\left(\sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2 + \left(z_2 - z_1\right)^2}\right)$

$R = \frac{1}{2}\left(\sqrt{\left(35 - 13\right)^2 + \left(40 - 15\right)^2 + \left(7 - 7\right)^2}\right)$

$R = 16.65$

➢ **Ellipse**

- Mathematically the ellipse is a curve generated by a point moving in space such that at any position the sum of its distances from two fixed points (foci) is constant and equal to the major diameter. Each focus is located on the major axis of the ellipse at a distance from its center equal to $\sqrt{A^2 - B^2}$ (A and B are the major and minor radii). Circular holes and forms become ellipses when they are viewed obliquely relative to their planes.

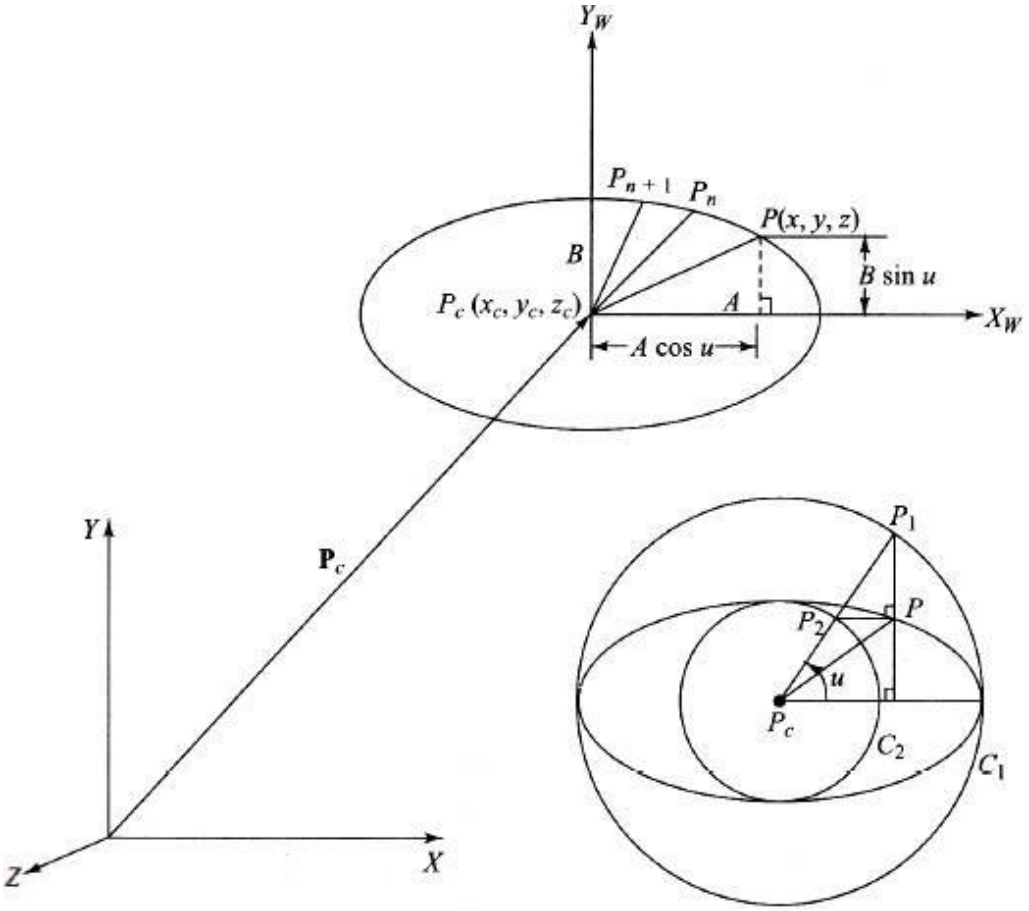


Fig. 2.8 Ellipse Defined by a Center, Major and Minor Axes

- However, four conditions (points and/or tangent vectors) are required to define the geometric shape of an ellipse.
- Fig. 2.8 shows an ellipse with point P_c as the center and the lengths of half of the major and minor axes are A and B respectively. The parametric equation of an ellipse can be written as assuming the plane of the ellipse is the XY plane.
- The parameter u is the angle as in the case of a circle. However, for a point P shown in the figure, it is not the angle between the line PP_c and the major axis of the ellipse. Instead, it is defined as shown.
- To find point P on the ellipse that corresponds to an angle u, the two concentric circles C_1 and C_2 are constructed with centers at P_c and radii of A and B respectively. A radial line is constructed at the angle u to intersect both circles at points P_1 and P_2 respectively.

- If a line parallel to the minor axis is drawn from P_1 and a line parallel to the major axis is drawn from P_2, the intersection of these two lines defines the point P.

$$\left.\begin{aligned} x &= x_c + A\cos u \\ y &= y_c + B\sin u \\ z &= z_c \end{aligned}\right\} \quad 0 \le u \le 2\pi$$

- Similar development as in the case of a circle results in the following recursive relationships which are useful for generating points on the ellipse for display purposes without excessive evaluations of trigonometric functions:

$$\begin{aligned} x_{n+1} &= x_c + (x_n - x_c)\cos\Delta u - \frac{A}{B}(y_n - y_c)\sin\Delta u \\ y_{n+1} &= y_c + (y_n - y_c)\cos\Delta u + \frac{B}{A}(x_n - x_c)\sin\Delta u \\ z_{n+1} &= z_n \end{aligned} \qquad (2.8)$$

- If the ellipse major axis is inclined with an angle α relative to the x axis as shown in Fig. 2.8, the ellipse equation becomes

$$\left.\begin{aligned} x &= x_c + A\cos u\cos\alpha - B\sin u\sin\alpha \\ y &= y_c + A\cos u\sin\alpha + B\sin u\cos\alpha \\ z &= z_c \end{aligned}\right\} \quad 0 \le u \le 2\pi \qquad (2.9)$$

- Equations (2.9) cannot be reduced to a recursive relationship similar to what is given by Eqs. (2.8). Instead these equations can be written as

$$\begin{aligned} x_{n+1} &= x_c + A\cos(u_n + \Delta u)\cos\alpha - B\sin(u_n + \Delta u)\sin\alpha \\ y_{n+1} &= y_c + A\cos(u_n + \Delta u)\sin\alpha + B\sin(u_n + \Delta u)\cos\alpha \\ z_{n+1} &= z_n \end{aligned}$$

➢ **Parabola**

- The parabola is defined mathematically as a curve generated by a point that moves such that its distance from a fixed point (the focus P_F) is always equal to its distance to a fixed line (the directrix) as shown in Fig. 2.9.
- The vertex P_v is the intersection point of the parabola with its axis of symmetry. It is located midway between directrix and focus. Focus lies on the axis of symmetry.
- Useful applications of the parabolic curve in engineering design include its use in parabolic sound and light reflectors, radar antennas and in bridge arches.
- Three conditions are required to define a parabola, a parabolic curve, or a parabolic arc. The default plane of a parabola is the XY plane of the current WCS at the time of construction.
- The database of a parabola usually stores the coordinates of its vertex, distances y_{HW} and y_{LW}, that define its endpoints as shown in Fig. 2.9, the distance A between the focus and the vertex (the focal distance) and the orientation angle α.
- Unlike the ellipse, the parabola is not a closed curve. Thus, the two endpoints determine the amount of the parabola to be displayed.

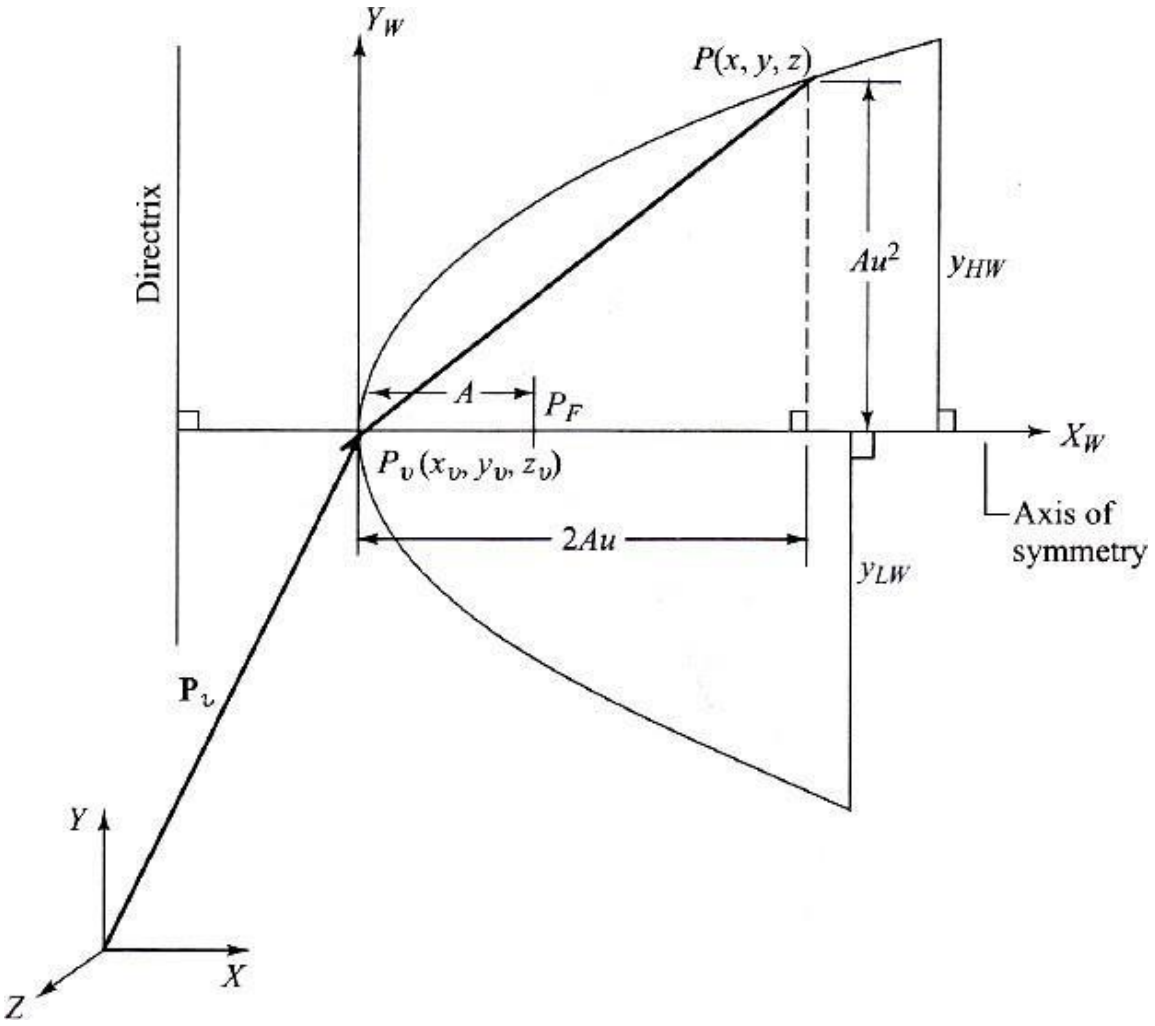


Fig. 2.9 Basic geometry of Parabola

- Assuming the local coordinate system of the parabola as shown in Fig. 2.9, its parametric equation can be written as

$$\left.\begin{aligned} x &= x_v + Au^2 \\ y &= y_v + 2Au \\ z &= z_v \end{aligned}\right\} \quad 0 \le u \le \infty$$

- If the range of the y coordinate is limited to y_{HW} and y_{LW} for positive and negative values respectively, the corresponding u values become

$$u_H = \frac{y_{HW}}{2A}$$

$$u_L = \frac{y_{LW}}{2A}$$

- The recursive relationships to generate points on the parabola are obtained by substituting $u_n + \Delta u$ for points n + 1. This gives

$$x_{n+1} = x_n + (y_n - y_v)\Delta u + A\,(\Delta u)^2$$
$$y_{n+1} = y_n + 2A\,\Delta u$$
$$z_{n+1} = z_n$$

➢ **Hyperbolas**

– A hyperbola is described mathematically as a curve generated by a point moving such that at any position the difference of its distances from the fixed points (foci) F and F' is a constant and equal to the transverse axis of the hyperbola.

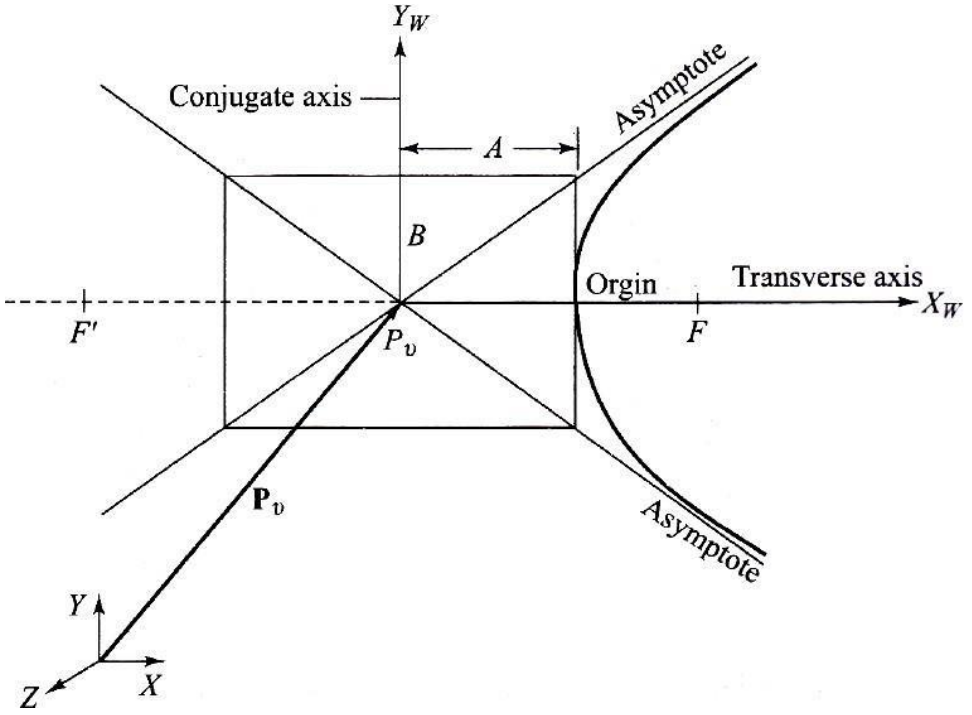


Fig. 2.10 Hyperbola Geometry

– Fig. 2.10 shows the geometry of a hyperbola.

– The parametric equation of a hyperbola is given by

$$x = x_v + A\cosh u$$
$$y = y_v + B\sinh u$$
$$z = z_v$$

– This equation is based on the nonparametric implicit equation of the hyperbola which can be written as

$$\frac{(x - x_v)^2}{A^2} - \frac{(y - y_v)^2}{B^2} = 1$$

by utilizing the identity $\cosh^2 u - \sinh^2 u = 1$.

Parametric representation of synthetic curve

- Analytic curves, are usually not sufficient to meet geometric design requirements of mechanical parts. Products such as car bodies, ship hulls, airplane fuselage and wings, propeller blades, shoe insoles and bottles are a few examples that require free-form, or synthetic, curves and surfaces.
- The need for synthetic curves in design arises on two occasions: when a curve is represented by a collection of measured data points and when an existing curve must change to meet new design requirements.
- In the latter occasion, the designer would need a curve representation that is directly related to the data points and is flexible enough to bend, twist, or change the curve shape by changing one or more data points.
- Data points are usually called control points and the curve itself is called an interpolant if it passes through all the data points.

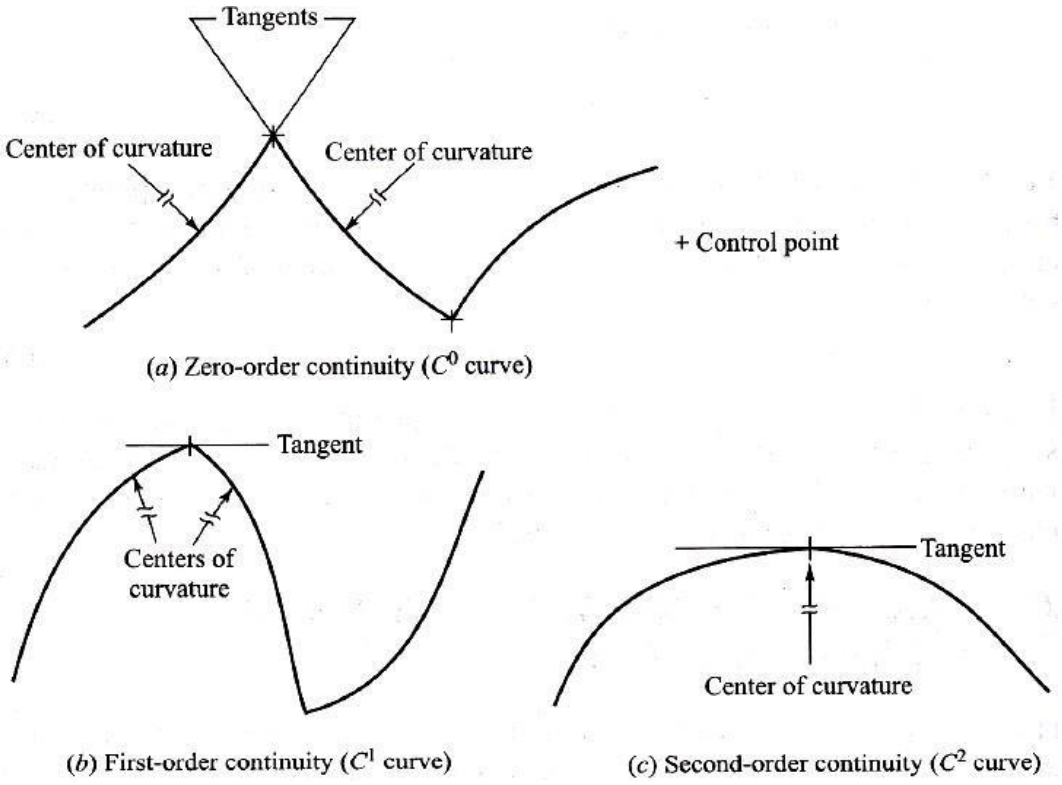


Fig. 2.11 Various Orders of Continuity of Curves

- Mathematically, synthetic curves represent a curve-fitting problem to construct a smooth curve that passes through given data points. Therefore, polynomials are the typical form of these curves.
- Various continuity requirements can be specified at the data points to impose various degrees of smoothness of the resulting curve. The order of continuity becomes important when a complex curve is modeled by several curve segments pieced together end to end.
- Zero-order continuity (C^0) yields a position continuous curve. First (C^1)- and second (C^2)-order continuities imply slope and curvature continuous curves respectively.

- A C^1 curve is the minimum acceptable curve for engineering design. Fig. 2.11 shows a geometrical interpretation of these orders of continuity.
- A cubic polynomial is the minimum order polynomial that can guarantee the generation of C^0, C^1, or C^2 curves. In addition, the cubic polynomial is the lowest-degree polynomial that permits inflection within a curve segment and that allows representation of nonplanar (twisted) three dimensional curves in space.
- Higher-order polynomials are not commonly used in CAD/CAM because they tend to oscillate about control points, are computationally inconvenient and are uneconomical of storing curve and surface representations in the computer.
- The type of input data and its influence on the control of the resulting synthetic curve determine the use and effectiveness of the curve in design.
- For example, curve segments that require positions of control points and/or tangent vectors at these points are easier to deal with and gather data for than those that might require curvature information.
- Also, the designer may prefer to control the shape of the curve locally instead of globally by changing the control point(s). If changing a control point results in changing the curve locally in the vicinity of that point, local control of the curve is achieved; otherwise global control results.
- Major CAD/CAM systems provide three types of synthetic curves: Hermite cubic spline, Bezier and B-spline curves. The cubic spline curve passes through the data points and therefore is an interpolant.
- Bezier and B-spline curves in general approximate the data points, that is, they do not pass through them. Under certain conditions, the B-spline curve can be an interpolant. Both the cubic spline and Bezier curves have a first-order continuity and the B-spline curve has a second-order continuity.

➢ **Hermite Cubic Spline Curve:**

- They are used to interpolate the given data but not to design free-form curves. Splines derive their name from “French curves or splines”
- It connects two data (end) points and utilizes a cubic equation.
- Four conditions are required to determine the coefficients of the equation – two end points and the two tangent vectors at these two points.
- It passes through the control points and therefore it is an Interpolant. It has only up to C_1 continuity.
- The curve cannot be modified locally, i.e., when a data point is moved, the entire curve is affected, resulting in a global control.
- The order of the curve is always constant (cubic), regardless of the number of data points. Increase in the number of data points increases shape flexibility, however, this requires more data points, creating more splines that are joined together (only two data points and slopes are utilized for each spline).

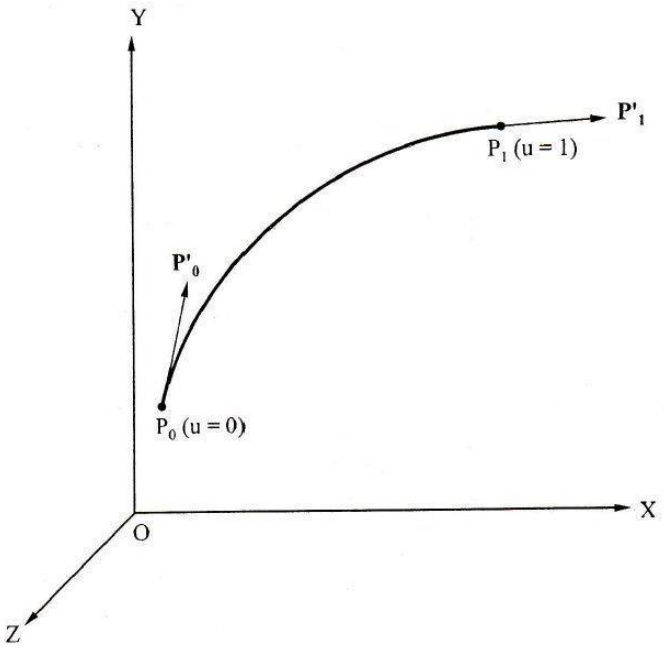


Fig. 2.12 Hermite Cubic Spline Curve

- A cubic spline curve utilizes a cubic equation of the following form:

$$P(u) = C_3 u^3 + C_2 u^2 + C_1 u + C_0 \quad (2.10)$$

where, $0 \le u \le 1$

- The cubic spline is defined by the two endpoints P_0, P_1 and the tangent vectors at these points. The tangent vector can be found by differentiating equ.(2.10).

$$P'(u) = 3C_3 u^2 + 2C_2 u + C_1 \quad (2.11)$$

- To determine the coefficients of the curve, consider the two endpoints P_0, P_1 and the tangent vectors as shown in Fig. 2.12.
- Applying conditions at $u = 0$, in equations (2.10) and (2.11)

$$P_0 = C_0 \quad (2.12)$$

$$P'_0 = C_1 \quad (2.13)$$

- Applying conditions at u = 1, in equations (2.10) and (2.11)

$$P_1 = C_3 + C_2 + C_1 + C_0 \quad (2.14)$$

$$P'_1 = 3C_3 + 2C_2 + C_1 \quad (2.15)$$

- Substituting (2.12) and (2.13) in (2.14) and (2.15), we get:

$$P_1 = C_3 + C_2 + P'_0 + P_0 \quad (2.16)$$

$$P'_1 = 3C_3 + 2C_2 + P'_0 \quad (2.17)$$

- By solving simultaneous equations (2.16) and (2.17), we get

$$3P_1 - P'_1 = C_2 + 2P'_0 + 3P_0$$

$$\therefore C_2 = 3P_1 - P'_1 - 2P'_0 - 3P_0 \quad (2.18)$$

- Similarly again by solving simultaneous equations (2.16) and (2.17), we get

$$P'_1 - 2P_1 = C_3 - P'_0 - 2P_0$$

$$\therefore C_3 = P'_1 - 2P_1 + P'_0 + 2P_0 \quad (2.19)$$

- Substituting (2.12), (2.13), (2.18) and (2.19) in (2.10), we get:

$$P(u)=\left(P'_1-2P_1+P'_0+2P_0\right)u^3+\left(3P_1-P'_1-2P'_0-3P_0\right)u^2+P'_0u+P_0$$

$$\therefore P(u)=P'_1u^3-2P_1u^3+P'_0u^3+2P_0u^3+3P_1u^2-P'_1u^2-2P'_0u^2-3P_0u^2+P'_0u+P_0$$

$$\therefore P(u)=2P_0u^3-3P_0u^2+P_0-2P_1u^3+3P_1u^2+P'_0u^3-2P'_0u^2+P'_0u+P'_1u^3-P'_1u^2$$

$$\therefore P(u)=P_0\left(2u^3-3u^2+1\right)+P_1\left(-2u^3+3u^2\right)+P'_0\left(u^3-2u^2+u\right)+P'_1\left(u^3-u^2\right) \qquad (2.20)$$

- The expression can be written in matrix form as under:

$$P(u)=\begin{bmatrix}P_0 & P_1 & P'_0 & P'_1\end{bmatrix}\begin{bmatrix}2u^3-3u^2+1\\-2u^3+3u^2\\u^3-2u^2+u\\u^3-u^2\end{bmatrix}$$

$$\therefore P(u)=\begin{bmatrix}P_0 & P_1 & P'_0 & P'_1\end{bmatrix}\begin{bmatrix}2&-3&0&1\\-2&3&0&0\\1&-2&1&0\\1&-1&0&0\end{bmatrix}\begin{bmatrix}u_2^3\\u\\u\\1\end{bmatrix}$$

- On similar lines, the tangent vector equation can be determined by differentiating equation (2.20)

$$P'(u)=P_0\left(6u^2-6u\right)+P_1\left(-6u^2+6u\right)+P'_0\left(3u^2-4u+1\right)+P'_1\left(3u^2-2u\right)$$

$$\therefore P'(u)=\begin{bmatrix}P_0 & P_1 & P'_0 & P'_1\end{bmatrix}\begin{bmatrix}6u^2-6u\\-6u^2+6u\\3u^2-4u+1\\3u^2-2u\end{bmatrix}$$

$$\therefore P'(u)=\begin{bmatrix}P_0 & P_1 & P'_0 & P'_1\end{bmatrix}\begin{bmatrix}0&6&-6&0\\0&-6&6&0\\0&3&-4&1\\0&3&-2&0\end{bmatrix}\begin{bmatrix}u_2^3\\u\\u\\1\end{bmatrix}$$

- Changing the values of endpoints or tangent vectors would modify the shape of the curve.

Example 2.5: A cubic spline curve has start point $P_0(16,0)$ and end point $P_1(3,1)$. The tangent vector for end point P_0 is given by line joining P_0 and point $P_2(14,8)$. Tangent vector for end P_1 is given by line joining P_2 and point P_1.

1. Determine the parametric equation of hermite cubic curve.
2. Plot the hermite cubic curve.

$P_0 = [16\ \ 0]$

$P_1 = [3\ \ 1]$

$P'_0 = P_2 - P_0 = [14\ 8] - [16\ \ 0]$

$= [-2\ 8]$

$P'_1 = P_1 - P_2 = [3\ 1] - [14\ \ 8]$

$= [-11\ -7]$

Parametric equation

For any point on curve

$$P(u)=\begin{bmatrix}P_0 & P_1 & P_0' & P_1'\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

X coordinate of a point on curve

$$P_x(u)=\begin{bmatrix}P_{0x} & P_{1x} & P_{0x}' & P_{1x}'\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_x(u)=\begin{bmatrix}16 & 3 & -2 & -11\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_x(u)=\begin{bmatrix}13 & -24 & -2 & 16\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_x(u) = 13u^3-24u^2-2u+16$$

Y coordinate of a point on curve

$$P_y(u) =\begin{bmatrix}P_{0y} & P_{1y} & P_{0y}' & P_{1y}'\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_y(u) =\begin{bmatrix}0 & 18 & -7\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_y(u)=\begin{bmatrix}-1 & -6 & 8 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_y(u)= -u^3-u^2+8u$$

Thus, parametric equation of parametric curve is

$$P_x(u) = 13u^3- 24u^2- 2u + 16$$
$$P_y(u)= -u^3- u^2+ 8u \qquad (2.21)$$

Curve plot

The point on curve obtained by varying value of u from 0 to 1 in steps of 0.1 in equ. (2.21) are shown in table

Point No.	0	1	2	3	4	5	6	7	8	9	10
U	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
x (u)	16	15.57	14.74	13.59	12.19	10.63	8.97	7.30	5.70	4.24	3.00
y (u)	0	0.74	1.35	1.83	2.18	2.38	2.42	2.32	2.05	1.61	1.00

The Hermite cubic curve plotted using the above coordinates is shown in Fig. 2.13.

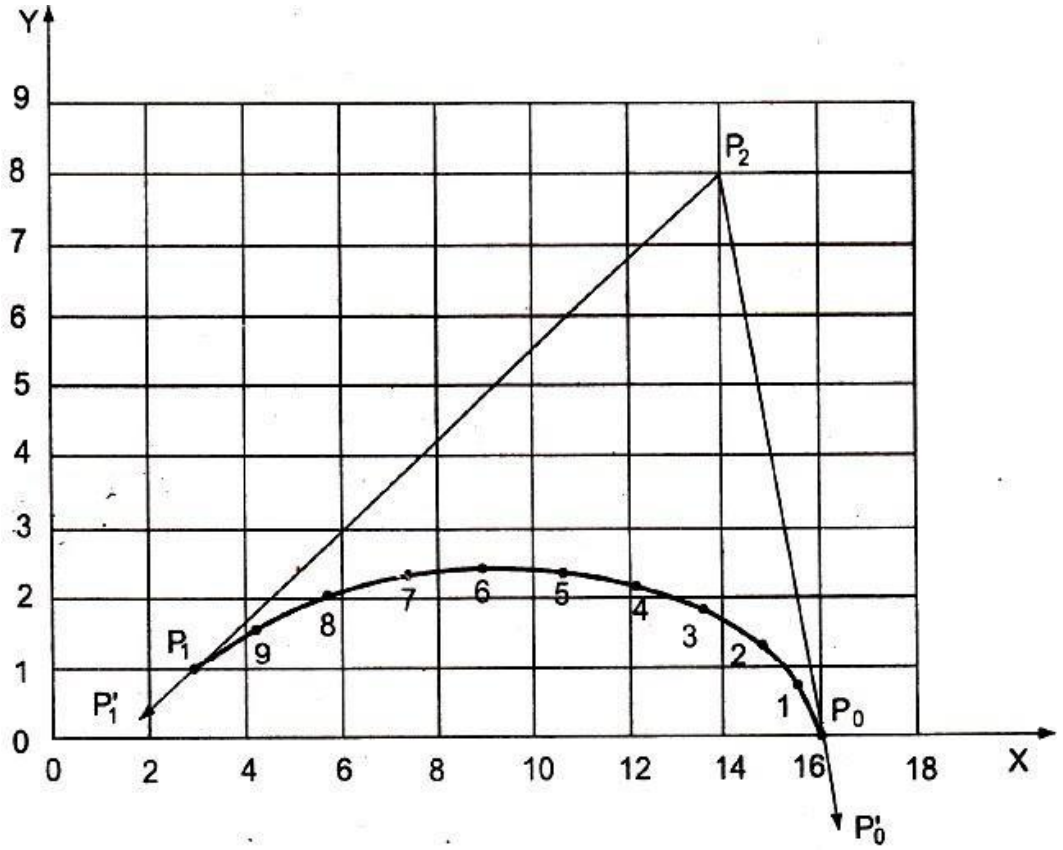


Fig. 2.13

Example 2.6: The end point of a cubic spline curve are $P_0(1,2)$ and $P_1(7,1)$. The tangent vector for end P_0 is given by line joining P_0 and point $P_2(-2,1)$. The tangent vector for end P_0 is given by line joining $P_3(9,-2)$ and point P_1.

1. Determine the parametric equation of hermite cubic curve.
2. Determine the parametric equation for tangent vector.
3. Plot the hermite cubic curve.

$P_0 = [1\ \ 2]$

$P_1 = [7\ \ 1]$

$P'_0 = P_2 – P_0 = [–2\ 1] – [1\ 2] = [–3\ –1]$

$P'_1 = P_1 – P_3 = [7\ 1] – [9\ –2] = [–2\ 3]$

Parametric equation of curve

For any point on curve

$$P(u)=\begin{bmatrix}P_0 & P_1 & P'_0 & P'_1\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

X coordinate of a point on curve

$$P_x(u)=\begin{bmatrix}P_{0x} & P_{1x} & P'_{0x} & P'_{1x}\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_x(u)=\begin{bmatrix}1 & 7 & -3 & -2\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_x(u)=\begin{bmatrix}-17 & 26 & -3 & 1\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_x(u)=-17u^3+26u^2-3u+1$$

Y coordinate of a point on curve

$$P_y(u)=\begin{bmatrix}P_{0y} & P_{1y} & P'_{0y} & P'_{1y}\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_y(u)=\begin{bmatrix}2 & 1 & -1 & 3\end{bmatrix}\begin{bmatrix}2 & -3 & 0 & 1\\ -2 & 3 & 0 & 0\\ 1 & -2 & 1 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_y(u)=\begin{bmatrix}4 & -4 & -1 & 2\end{bmatrix}\begin{bmatrix}u^3\\ u^2\\ u\\ 1\end{bmatrix}$$

$$P_y(u)=4u^3-4u^2-u+2$$

Thus, parametric equation of parametric curve is

$$P_x(u)=-17u^3+26u^2-3u+1$$
$$P_y(u)=4u^3-4u^2-u+2 \qquad (2.22)$$

Parametric equation of tangent vector

$$P'(u)=\begin{bmatrix} P_0 & P_1 & P'_0 & P'_1 \end{bmatrix}\begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

X coordinate of tangent vector

$$P'_x(u)=\begin{bmatrix} 1 & 7 & -3 & -2 \end{bmatrix}\begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_x(u)=\begin{bmatrix} 0 & 51 & 52 & -3 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_x(u)=51u^2+52u-3$$

Y coordinate of tangent vector

$$P'_y(u)=\begin{bmatrix} 2 & 1 & -1 & 3 \end{bmatrix}\begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_y(u)=\begin{bmatrix} 0 & 12 & -8 & -1 \end{bmatrix}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

$$P'_y(u)=12u^2-8u-1$$

Thus, parametric equation of tangent vector is

$$P'_x(u)=51u^2+52u-3$$

$$P'_y(u)=12u^2-8u-1$$

Curve plot

The point on curve obtained by varying the value of u from 0 to 1 in steps of 0.1 in equ.(2.22) are shown in table

Point No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
U	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
x (u)	1	0.94	1.30	1.98	2.87	3.88	4.89	5.81	6.54	6.97	7.00
y (u)	2	1.86	1.67	1.45	1.22	1.00	0.82	0.71	0.69	0.78	1.00

The Hermite cubic curve plotted using the above coordinates is shown in Fig. 2.14.

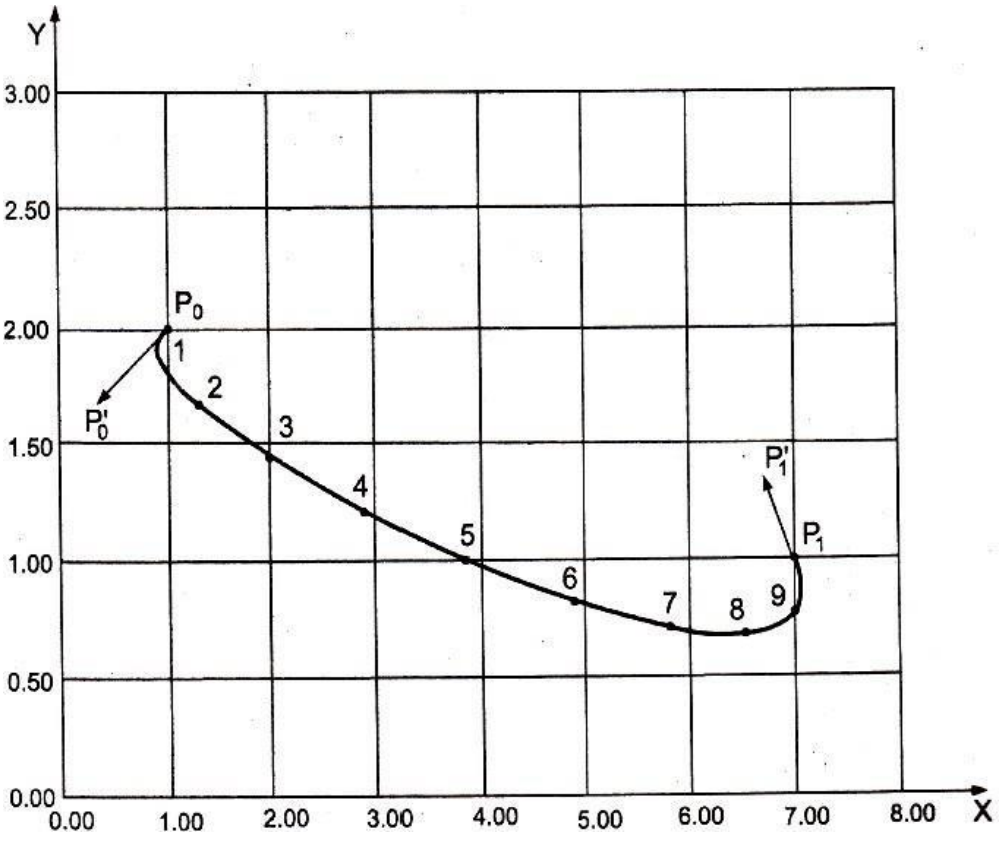


Fig. 2.14

- **Bezier Curves:**

- Cubic splines are based on interpolation techniques. Curves resulting from these techniques pass through the given points.
- Another alternative to create curves is to use approximation techniques which produce curves that do not pass through the given data points. Instead, these points are used to control the shape of the resulting curves.
- Most often, approximation techniques are preferred over interpolation techniques in curve design due to the added flexibility and the additional intuitive feel provided by the former. Bezier and B-spline curves are examples based on approximation techniques.
- As its mathematics show shortly, the major differences between the Bezier curve and the cubic spline curve are:

1. The shape of Bezier curve is controlled by its defining points only. First derivatives are not used in the curve development as in the case of the cubic spline. This allows the designer a much better feel for the relationship between input (points) and output (curve).
2. The order or the degree of Bezier curve is variable and is related to the number of points defining it; n + 1 points define an nth degree curve which permits higher-order continuity. This is not the case for cubic splines where the degree is always cubic for a spline segment.

3. The Bezier curve is smoother than the cubic spline because it has higher order derivatives.

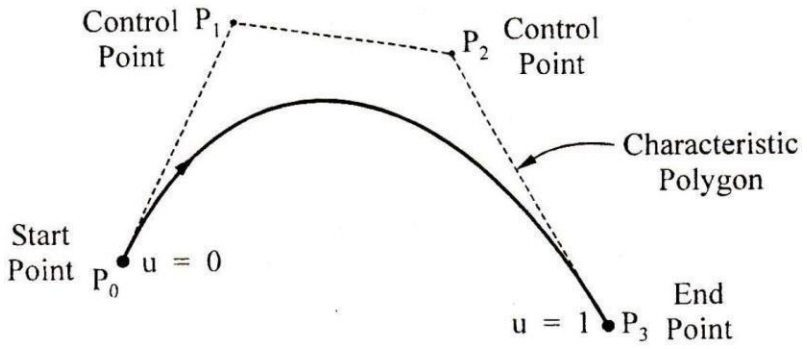


Fig. 2.15 Beizer Curve

- The Bezier curve is defined in terms of (n+1) points. This points are called as control points, where n is the degree of the curve.
- If n=3 control points are 4, as shown in the Fig 2.15.
- These control points form the vertices of the control polygon or Bezier characteristic polygon.
- The control or Bezier characteristic polygon uniquely defines the curve.
  - Properties of Beizer Curve:
  - The degree of polynomial defining the curve segment is one less than the no. of defining polygon points (n-1).
  - The curve follows the shape of the defining polygon.
  - Only the first and last control points or vertices of polygon actually lie on curve.
  - The other vertices define order & shape of the curve. (see Fig. 3.8)

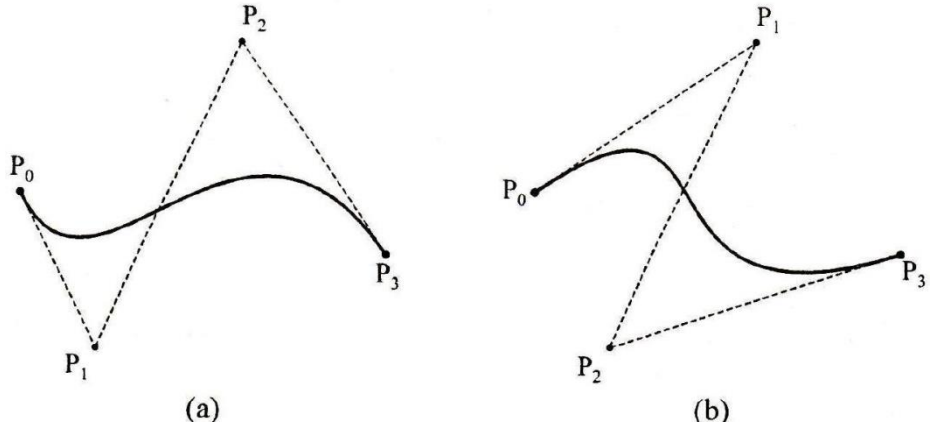


Fig. 2.16 Same data points but different orders for Beizer Curve

  - The curve is also always tangent to first and last segments of polygon.
  - The same Beizer curve would be generated, if the sequence of the points is changed from $P_0 – P_1 – P_2 – P_3$ to $P_3 – P_2 – P_1 – P_0$.
  - The flexibility of the shape would increase with increase in number of vertices of the polygon.
  - It is having global control on the shape of the curve.

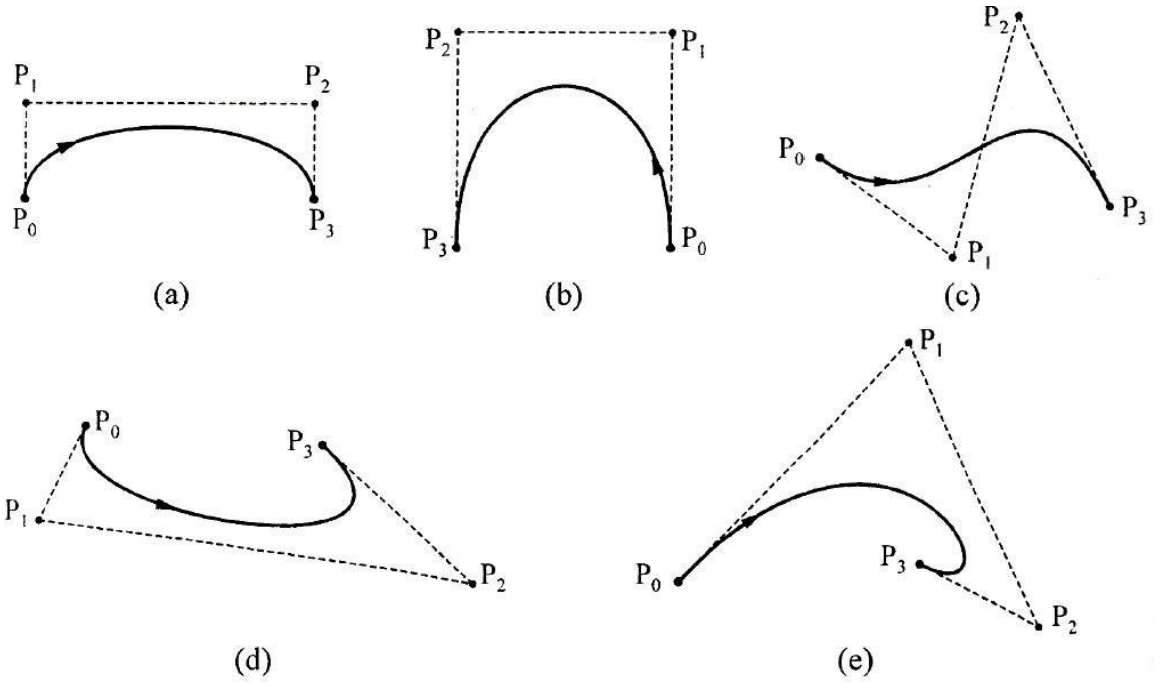


Fig. 2.17 Same Typical Examples of Beizer Curve

- Mathematically, for n + 1 control points, the Bezier curve is defined by the following polynomial of degree n:

$$P(u) = \sum_{i=0}^{n} P_i B_{i,n}(u), \quad 0 \le u \le 1 \tag{2.23}$$

- where P(u) is any point on the curve and P_i is a control point. $B_{i,n}$ are the Bernstein polynomials. Thus, the Bezier curve has a Bernstein basis. The Bernstein polynomial serves as the blending or basis function for the Bezier curve and is given by

$$B_{i,n}(u) = C(n,i)u^i(1-u)^{n-i} \tag{2.24}$$

- where C(n,i) is the binomial coefficient

$$C(n,i) = \frac{n!}{i!(n-i)!} \tag{2.25}$$

- Utilizing Eqs. (2.24) and (2.25) and observing that C(n, 0) = C(n, n) = 1,
- Eq. (2.23) can be expanded to give

$$P(u) = P_0(1-u)^n + P_1 C(n,1)u(1-u)^{n-1} + P_2 C(n,2)u^2(1-u)^{n-2} + \ldots + P_{n-1} C(n,n-1)u^{n-1}(1-u) + P_n u^n, \quad 0 \le u \le 1$$

- Bezier curve with 3 control points

$$P(u) = P_0(1-u)^2 + P_1 C(2,1)u(1-u)^{2-1} + P_2 C(2,2)u^2(1-u)^{2-2}$$
$$P(u) = (1-u)^2 P_0 + 2u(1-u)P_1 + u^2 P_2$$

- Bezier curve with 4 vertices

$$P(u) = P_0(1-u)^3 + P_1 C(3,1)u(1-u)^{3-1} + P_2 C(3,2)u^2(1-u)^{3-2} + P_3 C(3,3)u^3(1-u)^{3-3}$$
$$P(u) = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u)P_2 + u^3 P_3$$

- Bezier curve with 5 vertices

$$P(u) = P_0(1-u)^4 + P_1 C(4,1)u(1-u)^{4-1} + P_2 C(4,2)u^2(1-u)^{4-2} + P_3 C(4,3)u^3(1-u)^{4-3} + P_4 C(4,4)u^4(1-u)^{4-4}$$

$$P(u) = (1-u)^4 P_0 + 4u(1-u)^3 P_1 + 6u^2(1-u)^2 P_2 + 4u^3(1-u) P_3 + u^4 P_4$$

Example 2.7: A Bezier curve is to be constructed using control points $P_0(35,30)$, $P_1(25,0)$, $P_2(15,25)$ and $P_3(5,10)$. The Bezier curve is anchored at P_0 and P_3. Find the equation of the Bezier curve and plot the curve for u = 0, 0.2, 0.4, 0.6, 0.8 and 1.

P_0 = [35 30]
P_1 = [25 0]
P_2 = [15 25]
P_3 = [5 10]
n = 3

The parametric equation for Bezier curve

$$P(u) = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u) P_2 + u^3 P_3$$

X-coordinate of a point on curve

$$P_x(u) = (1-u)^3 P_{0x} + 3u(1-u)^2 P_{1x} + 3u^2(1-u) P_{2x} + u^3 P_{3x}$$

$$P_x(u) = 35(1-u)^3 + 75u(1-u)^2 + 45u^2(1-u) + 5u^3$$

$$P_x(u) = 35(1 - 3u + 3u^2 - u^3) + 75u(1 - 2u + u^2) + 45u^2 - 45u^3 + 5u^3$$

$$P_x(u) = 35 - 105u + 105u^2 - 35u^3 + 75u - 150u^2 + 75u^3 + 45u^2 - 45u^3 + 5u^3$$

$$P_x(u) = 35 - 30u$$

Y-coordinate of a point on curve

$$P_y(u) = (1-u)^3 P_{0y} + 3u(1-u)^2 P_{1y} + 3u^2(1-u) P_{2y} + u^3 P_{3y}$$

$$P_y(u) = 30(1-u)^3 + 75u^2(1-u) + 10u^3$$

$$P_y(u) = 30(1 - 3u + 3u^2 - u^3) + 75u^2 - 75u^3 + 10u^3$$

$$P_y(u) = 30 - 90u + 165u^2 - 95u^3$$

The parametric equation of Bezier curve

$$P_x(u) = 35 - 30u$$

$$P_y(u) = 30 - 90u + 165u^2 - 95u^3 \qquad (2.26)$$

The points on curve obtained by varying the value of u = 0, 0.2, 0.4, 0.6, 0.8 and 1 in equ. (2.26) are shown in below table

Point No.	**0**	**2**	**4**	**6**	**8**	**10**
u	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**
P_x (u)	35.0	29.0	23.0	17.0	11.0	5.00
P_y (u)	30.0	17.84	14.32	14.88	14.96	10.00

The Bezier curve plotted using the above coordinates is shown in Fig. 2.18.

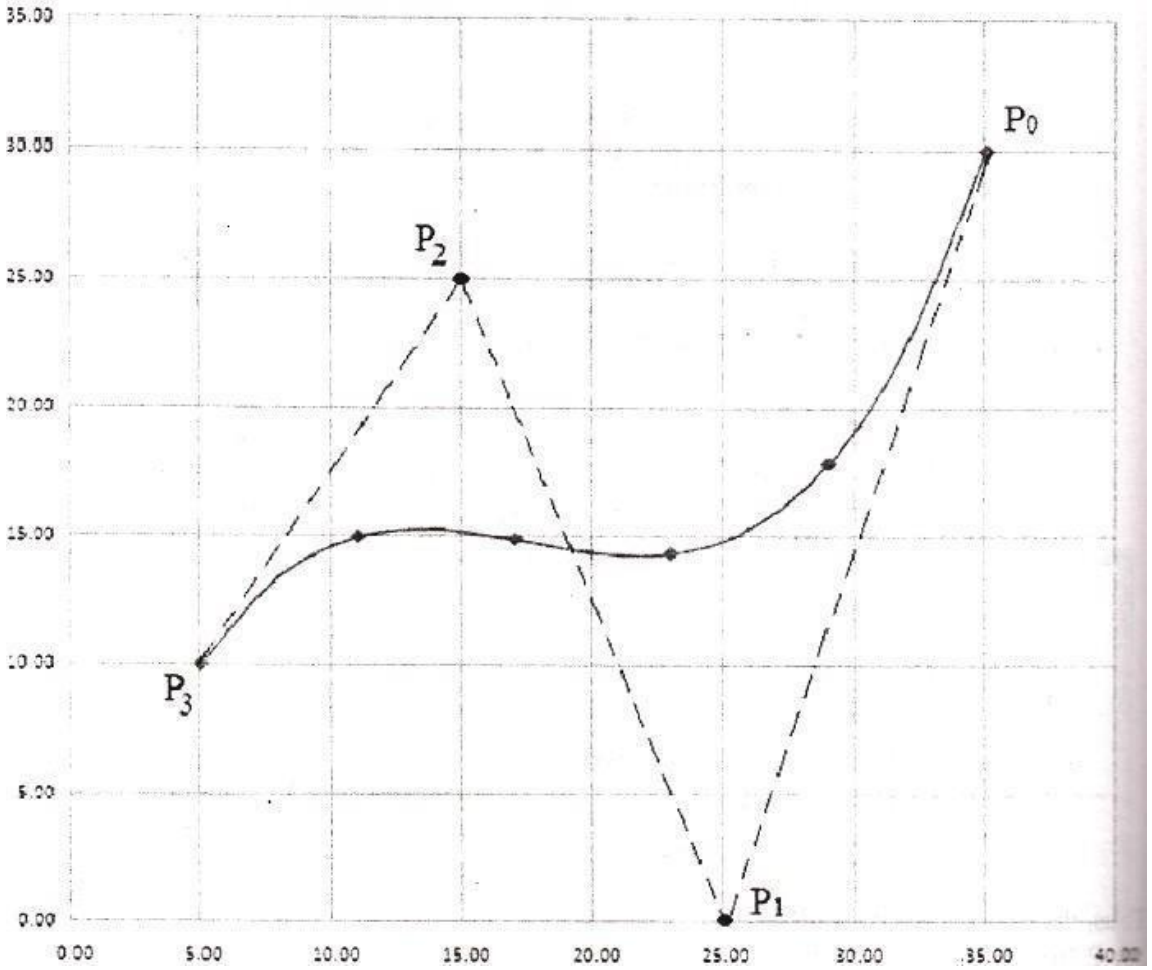


Fig. 2.18

Example 2.8: Plot a Bezier curve using the following control points.
(2,0), (4,3), (5,2), (4,-2), (5,-3) and (6,-2).

$P_0 = [2\ 0]$ $P_1 = [4\ \ 3]$
$P_2 = [5\ 2]$ $P_3 = [4\ \ -2]$
$P_4 = [5\ -3]$ $P_5 = [6\ \ -2]$
$n = 5$

The parametric equation for Bezier curve with 6 control points

$$P(u) = (1-u)^5 P_0 + 5u(1-u)^4 P_1 + 10u^2(1-u)^3 P_2 + 10u^3(1-u)^2 P_3 + 5u^4(1-u) P_4 + u^5 P_5$$

$$P(u) = (1 - 5u + 10u^2 - 10u^3 + 5u^4 - u^5) P_0 + 5u(1 - 4u + 6u^2 - 4u^3 + u^4) P_1$$
$$+ 10u^2(1-3u+3u^2-u^3) P_2 + 10u^3(1-2u+u^2) P_3 + 5u^4(1-u) P_4 + u^5 P_5$$

$$P(u) = (1 - 5u + 10u^2 - 10u^3 + 5u^4 - u^5) P_0 + (5u - 20u^2 + 30u^3 - 20u^4 + 5u^5) P_1$$
$$+ (10u^2 - 30u^3 + 30u^4 - 10u^5) P_2 + (10u^3 - 20u^4 + 10u^5) P_3 + (5u^4 - 5u^5) P_4 + u^5 P_5$$

$$P(u) = P_0 + (-5P_0 + 5P_1)u + (10P_0 - 20P_1 + 10P_2)u^2 + (-10P_0 + 30P_1 - 30P_2 + 10P_3)u^3$$
$$+ (5P_0 - 20P_1 + 30P_2 - 20P_3 + 5P_4)u^4 + (-P_0 + 5P_1 - 10P_2 + 10P_3 - 5P_4 + P_5)u^5$$

X-coordinate of a point on a curve

$$P_x(u) = P_{0x} + (-5P_{0x} + 5P_{1x})u + (10P_{0x} - 20P_{1x} + 10P_{2x})u^2 + (-10P_{0x} + 30P_{1x} - 30P_{2x} + 10P_{3x})u^3$$
$$+ (5P_{0x} - 20P_{1x} + 30P_{2x} - 20P_{3x} + 5P_{4x})u^4 + (-P_{0x} + 5P_{1x} - 10P_{2x} + 10P_{3x} - 5P_{4x} + P_{5x})u^5$$

$$P_x(u) = 2 + (-5(2) + 5(4))u + (10(2) - 20(4) + 10(5))u^2 + (-10(2) + 30(4) - 30(5) + 10(4))u^3$$
$$+ (5(2) - 20(4) + 30(5) - 20(4) + 5(5))u^4 + (-(2) + 5(4) - 10(5) + 10(4) - 5(5) + (6))u^5$$

$$P_x(u) = 2 + 10u - 10u^2 - 10u^3 + 25u^4 - 11u^5$$

Y-coordinate of a point on a curve

$$P_y(u) = P_{0y} + (-5P_{0y} + 5P_{1y})u + (10P_{0y} - 20P_{1y} + 10P_{2y})u^2 + (-10P_{0y} + 30P_{1y} - 30P_{2y} + 10P_{3y})u^3$$
$$+ (5P_{0y} - 20P_{1y} + 30P_{2y} - 20P_{3y} + 5P_{4y})u^4 + (-P_{0y} + 5P_{1y} - 10P_{2y} + 10P_{3y} - 5P_{4y} + P_{5y})u^5$$

$$P_y(u) = 0 + (-5(0) + 5(3))u + (10(0) - 20(3) + 10(2))u^2 + (-10(0) + 30(3) - 30(2) + 10(-2))u^3$$
$$+ (5(0) - 20(3) + 30(2) - 20(-2) + 5(-3))u^4 + (-(0) + 5(3) - 10(2) + 10(2) - 5(-3) + (-2))u^5$$

$$P_y(u) = 15u - 40u^2 + 10u^3 + 25u^4 - 12u^5$$

The parametric equation of Bezier curve

$$P_x(u) = 2 + 10u - 10u^2 - 10u^3 + 25u^4 - 11u^5 \qquad (2.27)$$
$$P_y(u) = 15u - 40u^2 + 10u^3 + 25u^4 - 12u^5$$

The points on curve obtained by varying the value of u from 0 to 1 in steps of 0.1 in equ.(2.27) are shown in below table

Point No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
u	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
P_x (u)	2	2.89	3.56	4.01	4.29	4.47	4.62	4.82	5.12	5.52	6.00
P_Y (u)	0	1.11	1.52	1.34	0.76	-0.06	-0.93	-1.68	-2.17	-2.29	-2.00

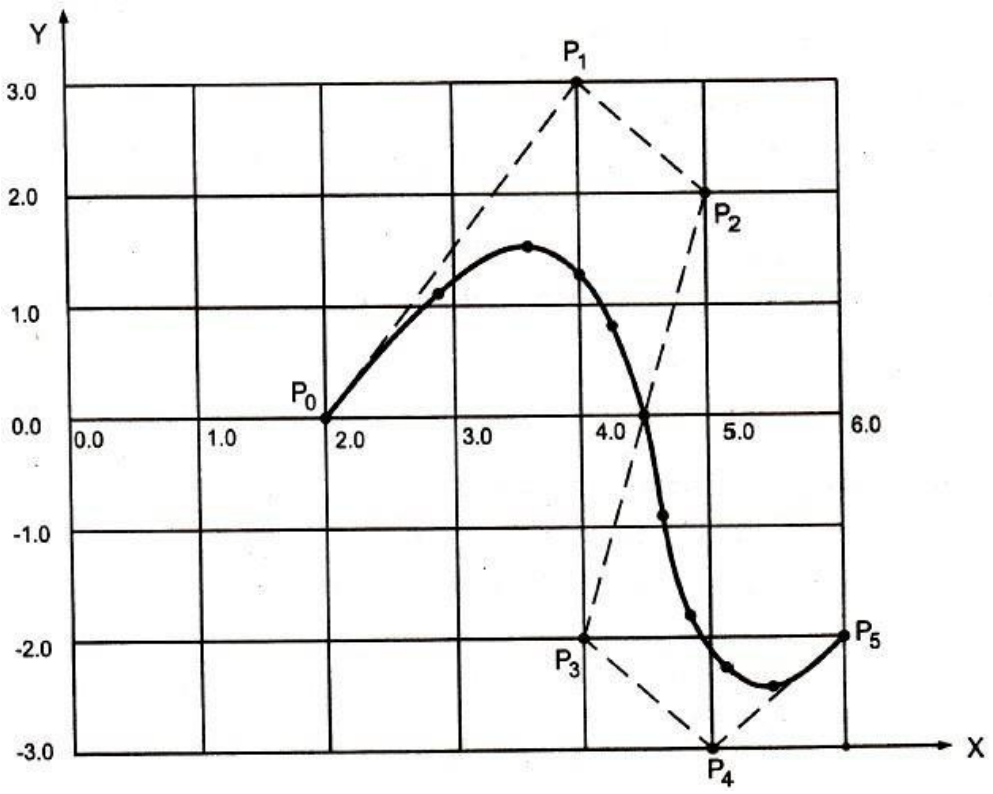


Fig. 2.19

➢ **B-Spline Curve:**

- B-Spline curves are proper and powerful generalization of Bezier curve.
- They are very similar to Bezier curve and are also defined with the help of characteristic polygon.
- The major advantage of B-Spline curve is due to its ability to control the curve shape locally, as opposed to global control (see Fig. 2.20).

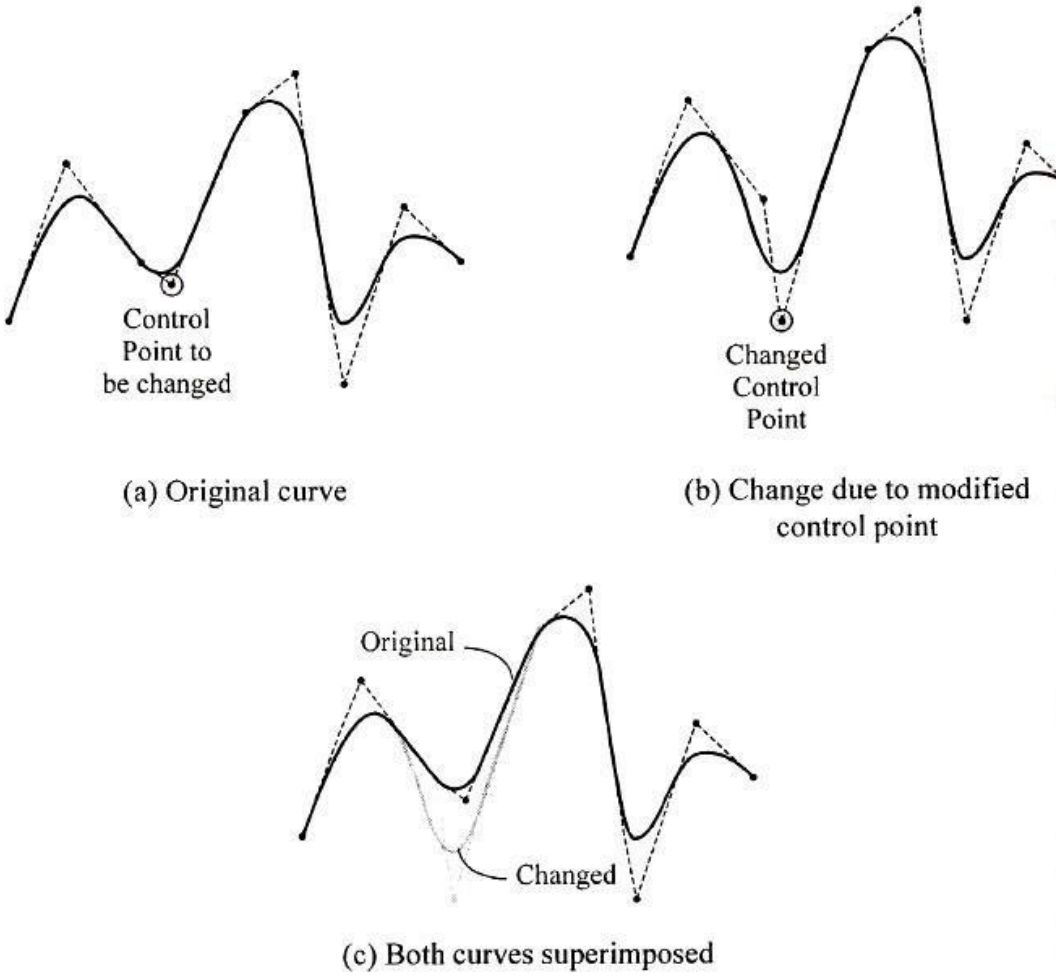


Fig. 2.20 Local Control of a B-Spline Curve

- B-Spline curves allow a local control over the shape of the spline curve.
- B-Spline curves allow us to vary the control points without changing the degree of the polynomial (i.e. number of control points can be added or subtracted).
- The degree of the curve is independent of the number of control points (i.e. four control points can generate a linear, quadratic or cubic curve).
- It gives better control over the shape of curve.
- They are more complex compared to Bezier curves.
- B-spline curve defined by n + 1 control points P_i is given by

$$P(u) = \sum_{i=0}^{n} P_i N_{i,k}(u), \quad 0 \le u \le u_{max} \tag{2.28}$$

$N_{i,k}(u)$ are the B-spline functions. Thus, the B-spline curves have a B-spline basis.

- **Rational curves**

- A rational curve is defined by the algebraic ratio of two polynomials while a nonrational curve is defined by one polynomial.
- The formulation of rational curves requires the introduction of homogeneous space and the homogeneous coordinates. The homogeneous space is four-dimensional space. A point in three dimensional with coordinates (x, y, z) is represented in the homogeneous space by the coordinates (x*, y*, z*, h),where h is a scalar factor. The relationship between the two types of coordinates is

$$x = \frac{x^*}{h} \qquad y = \frac{y^*}{h} \qquad z = \frac{z^*}{h}$$

- A rational B-spline curve defined by n + 1 control points P_i is given by

$$P(u) = \sum_{i=0}^{n} P_i R_{i,k}(u), \quad 0 \le u \le u_{max} \tag{2.29}$$

- $R_{i,k}(u)$ are the rational B-spline functions and are given by

$$R_{i,k}(u) = \frac{h_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)}$$

- The above equation shows that $R_{i,k}(u)$ are a generalization of the nonrational basis functions $N_{i,k}(u)$. If we substitute $h_i = 1$ in the equation, $R_{i,k}(u) = N_{i,k}(u)$. The rational basis functions $R_{i,k}(u)$ have nearly all the analytic and geometric characteristics of their nonrational B-spline counterparts.
- The main difference between rational and nonrational B-spline curves is the ability to use h_i at each control point to control the behavior of the rational B-splines (or rational curves in general).
- Thus, similarly to the knot vector, one can define a homogeneous coordinate vector $H = [h_0\ h_1\ h_2\ h_3\ \ldots\ h_n]^T$ at the control points $P_0, P_1, \ldots, P_n$ of the rational B-spline curve. The choice of the H vector controls the behavior of the curve.
- A rational B-spline is considered a unified representation that can define a variety of curves and surfaces. The premise is that it can represent all wireframe, surface and solid entities. This allows unification and conversion from one modeling technique to another. Such an approach has some drawbacks, including the loss of information on simple shapes.
- For example, if a circular cylinder (hole) is represented by a B-spline, some data on the specific curve type may be lost unless it is carried along. Data including the fact that the part feature was a cylinder would be useful to manufacturing to identify it as a hole to be drilled or bored rather than a surface to be milled.

Surface Modeling

- The surface model is an extension of wire frame model which involves providing a surface representation making the object look solid to the viewer.
- Surface systems were in fact the first CAD systems to raise the possibility of integrating the whole industrial process of design and analysis through to the next stages of production and quality control, using the computer as an intermediary.
- Shape design and representation of complex objects such as car, ship, aircraft and castings cannot be achieved by a wire frame model. In such cases, surface models are preferred for representing the objects with precision and accuracy.
- These models also find widespread applications in companies manufacturing forgings, castings and molded products. Such articles are characterized by smoothly curved shapes with blended edges which are not easily represented by conventional engineering drawings.
- Another difficulty for designers with wire frame modelers is to obtain good visualization of their ideas. The surface modeler is ideal for such products as the entire surface is represented; allowing the computer to generate very realistic shaded surface pictures.
- Most surface modelers come equipped with *rendering* features. In this, the model once created is then provided with surface properties. For example, the surface may be given a property which may make the object appear corroded or made of brass or any such effects.
- A huge material library is generally made available to select the desired surface effect. One can also define different kinds of lights, such as spot lights, ambient lights, etc. in a 3D space and generate a photorealistic image of the object. This feature is highly used in automobile and styling industries for determining the aesthetic appeal of the object being designed.
- Advanced surface modelers go beyond representation and to some extent can be used for property calculations as well. The surface model can be used for generating NC tool paths for continuous path machining, making it an ideal mode for integrating CAD / CAM. A surface modeler can be considered as an extension of a wire frame.
- A wire frame model can be easily extracted from a surface model. It should be remembered that surface models only store the geometry of their corresponding objects and not the topology of these objects. To generate a surface model, a user typically starts with a wire frame model and then connects them with the desired surfaces.
- Once the product has been surface modeled, its geometry may be used directly in the design of the mould or die which is to be made. Essential calculations can usually be performed automatically by the computer such as for example the determination of the volume enclosed by a mould. This gives an immediate idea of the volume of raw material needed to make the product.

- **Kinds of Surfaces**

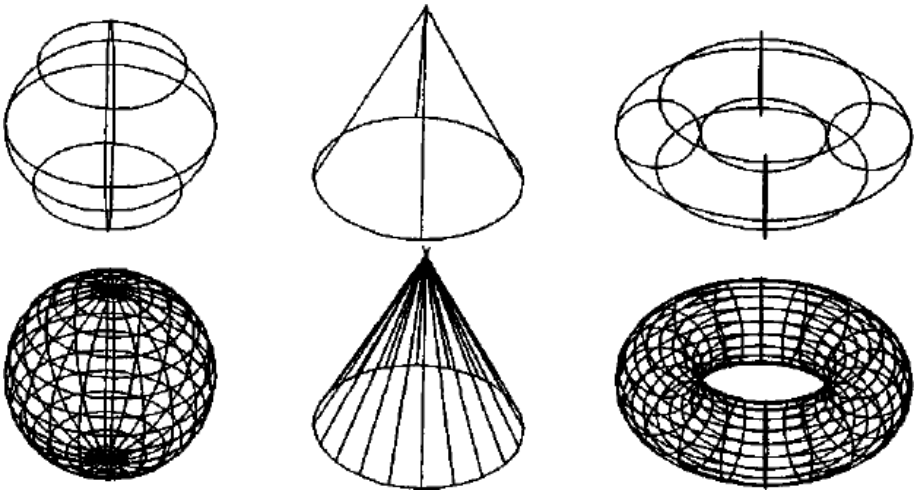


Fig. 2.21 Effect of Mesh Size on Object Visualization

- To create a surface, data related to the desired shape is required. The choice of the surface is dependent on the application. In order to visualize surfaces, a *mesh* of *m* x *n* in size is displayed. The mesh size is controllable and is in the form of criss-cross on the surface. The mesh size plays an important role in proper visualization as can be seen in the objects shown in *Fig. 2.21*. However, it should be noted that finer the mesh size of surface entities, longer is the time required by the software to construct and modify the entities.
- A system may need two boundaries to create a ruled surface or might require one entity to create a surface of revolution. Surfaces provided by CAD / CAM software are either *analytical* or *synthetic.*

The following are descriptions of major surface entities provided by CAD/CAM systems:

1. **Plane Surface:** This is the simplest surface. It requires three noncoincident points to define an infinite plane. The plane surface can be used to generate cross-sectional views by intersecting a surface model with it, generate cross sections for mass property calculations, or other similar applications where a plane is needed. Fig. 2.22 shows a plane surface.

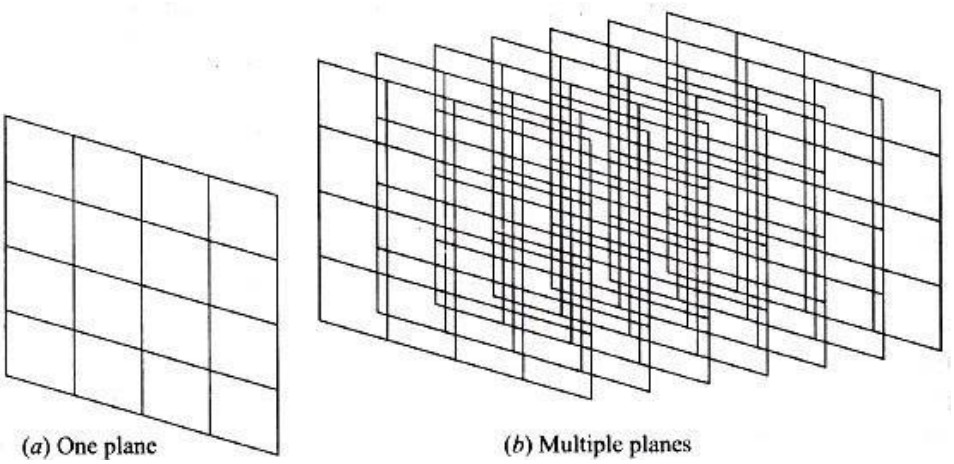


Fig. 2.22 Plane Surface

2. **Ruled (lofted) Surface:** This is a linear surface. It interpolates linearly between two boundary curves that define the surface (rails). Rails can be any wireframe entity. This entity is ideal to represent surfaces that do not have any twists or kinks. Fig. 2.23 gives some examples.

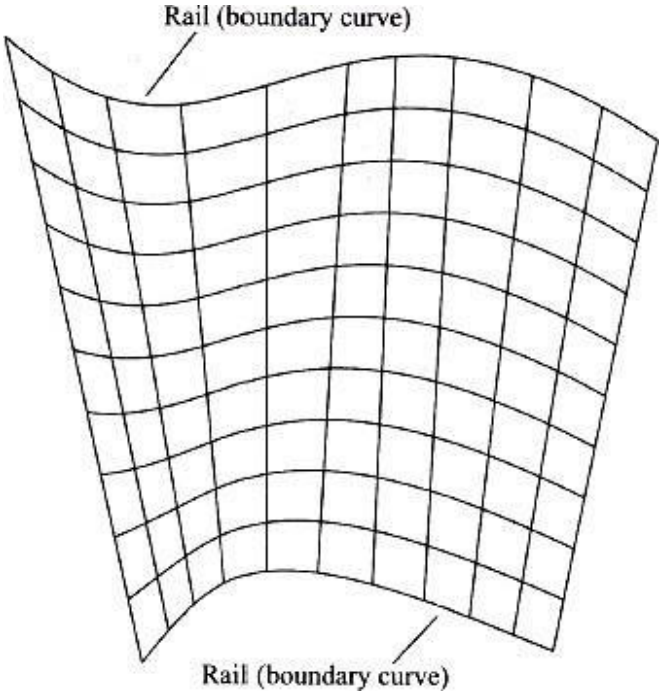


Fig. 2.23 Ruled Surface

3. **Surface of Revolution:** This is an axisymmetric surface that can model axisymmetric objects. It is generated by rotating a planar wireframe entity in space about the axis of symmetry a certain angle (Fig. 2.24).

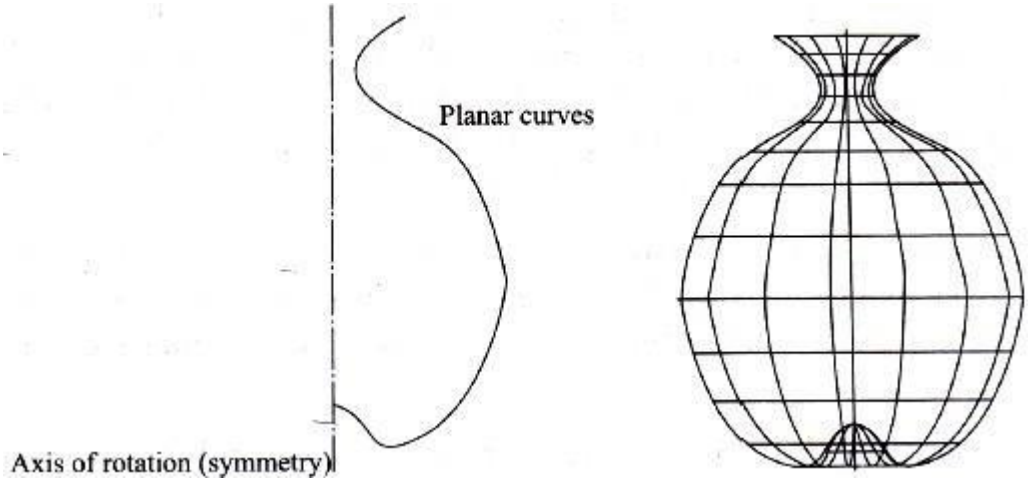


Fig. 2.24 Surface of Revolution

4. **Tabulated Cylinder:** This is a surface generated by translating a planar curve a certain distance along a specified direction (axis of the cylinder) as shown in Fig. 2.25. The plane of the curve is perpendicular to the axis of the cylinder. It is used to generate surfaces that have identical curved cross sections. The word "tabulated" is borrowed from the APT language terminology.

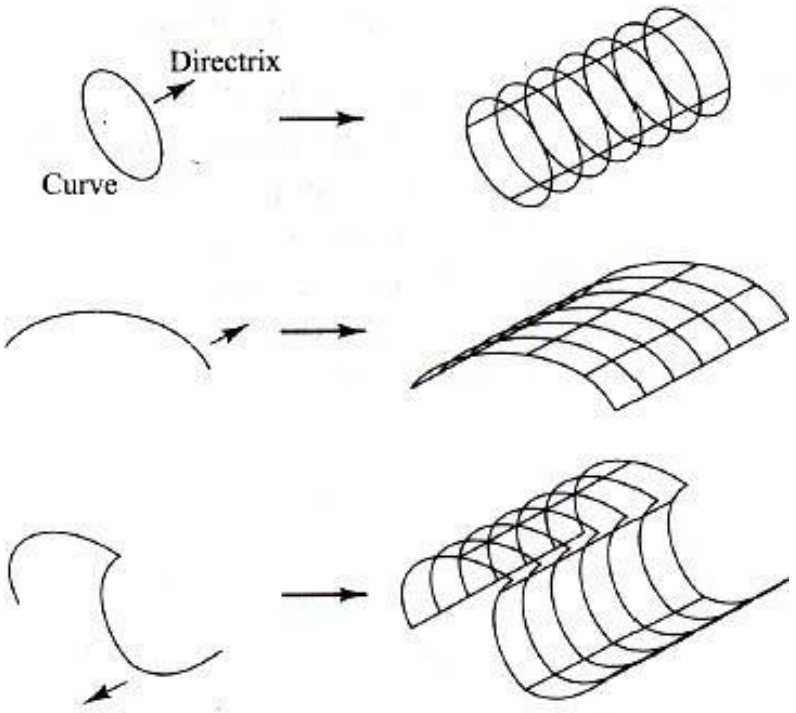


Fig. 2.25 Tabulated Cylinder

5. **Bezier surface:** This is a surface that approximates given input data. It is different from the previous surfaces in that it is a synthetic surface. Similarly to the Bezier curve, it does not pass through all given data points. It is a general surface that permits twists and kinks (Fig. 2.26). The Bezier surface allows only global control of the surface.

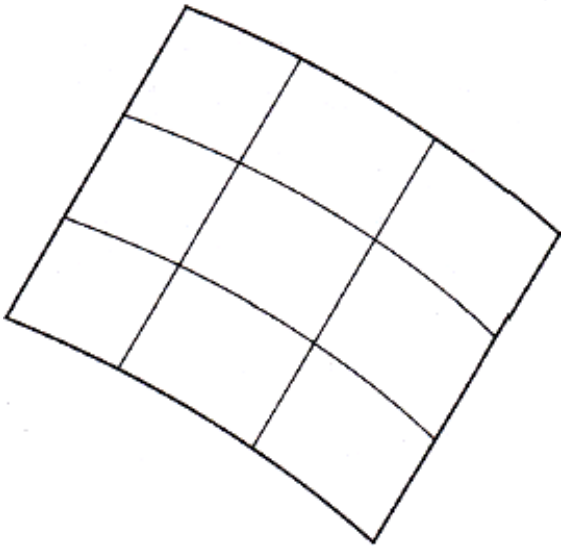


Fig. 2.26 Bezier Surface

6. **B-spline surface:** This is a surface that can approximate or interpolate given input data (Fig. 2.27). It is a synthetic surface. It is a general surface like the Bezier surface but with the advantage of permitting local control of the surface.

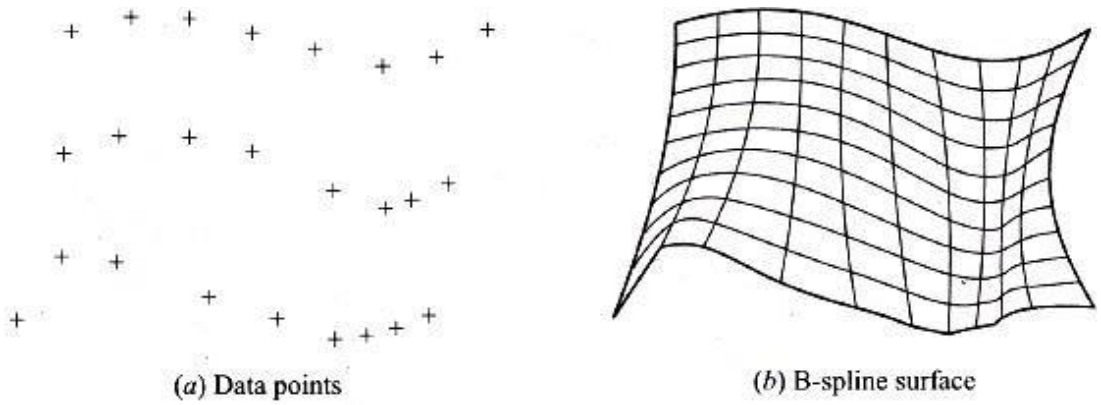


Fig. 2.27 B-spline Surface

7. **Coons Patch:** The above surfaces are used with either open boundaries or given data points. The Coons patch is used to create a surface using curves that form closed boundaries (Fig. 2.28).

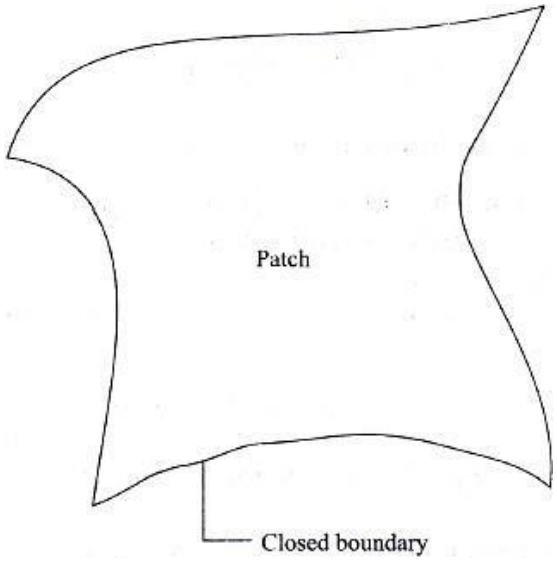


Fig. 2.28 Coons Patch

8. **Fillet surface:** This is a B-spline surface that blends two surfaces together (Fig. 2.29). The two original surfaces may or may not be trimmed.

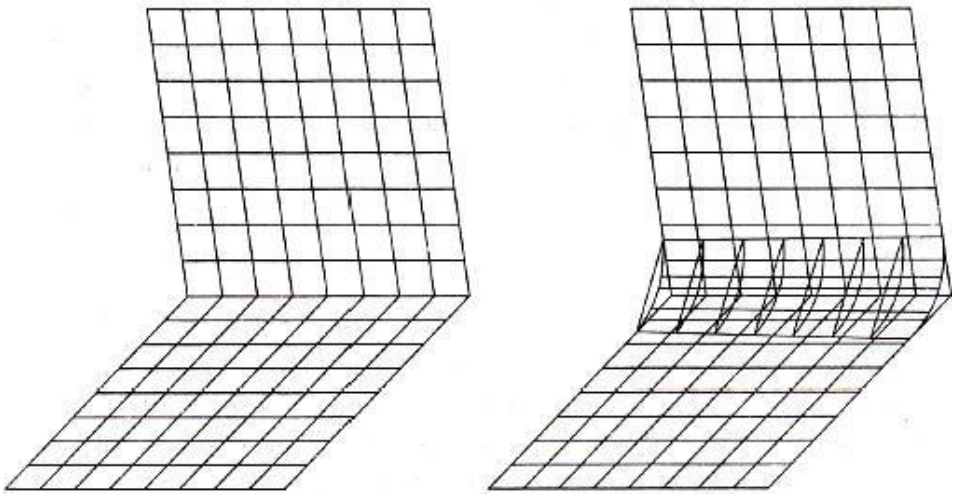


Fig. 2.29 Fillet Surface

9. **Offset surface:** Existing surfaces can be offset to create new ones identical in shape but may have different dimensions. It is a useful surface to use to speed up surface construction. For example, to create a hollow cylinder, the outer or inner cylinder can be created using a cylinder command and the other one can be created by an offset command. Offset surface command becomes very efficient to use if the original surface is a composite one. Figure 2.30 shows an offset surface.

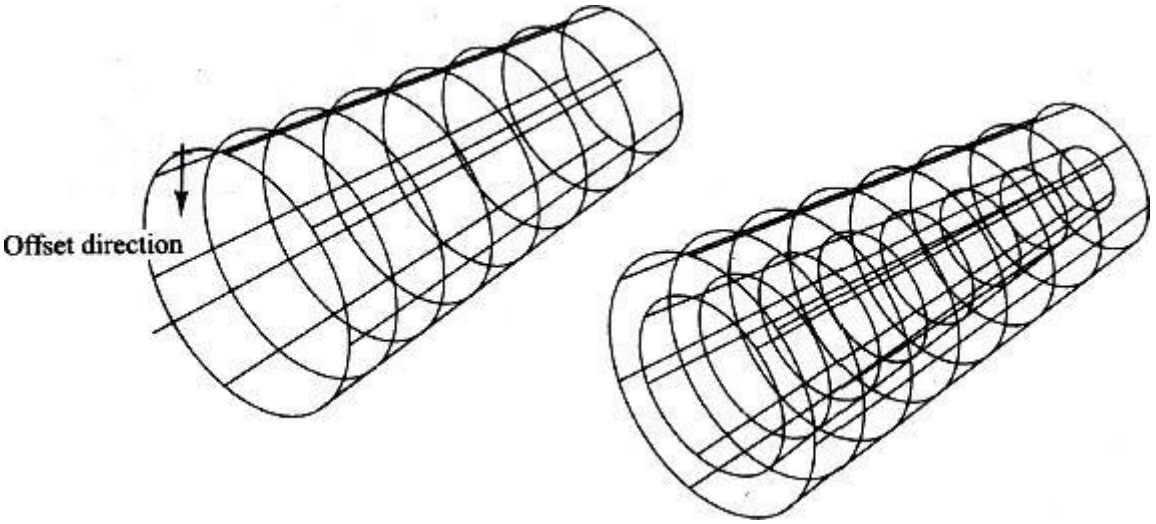


Fig. 2.30 Offset Surface

Tutorial Questions

1. Explain representation of lines at different cases.
2. Explain representation of circle, ellipse.
3. Explain representation of parabola and hyperbola.
4. Explain properties, equation and blending of cubic spline.
5. Explain properties, equation and blending of Bezier curves.
6. Explain properties, and advantages of B-Splines and NURBS.
7. Explain various types of surfaces.

QUESTION BANK

1. Explain representation of lines at different cases.
2. Explain properties, equation and blending of cubic spline.
3. Explain properties, equation and blending of Bezier curves.
4. Explain properties, and advantages of B-Splines and NURBS.
5. Explain various types of surfaces.

UNIT 3

MATHEMATICAL REPRESENTATION OF SOLIDS

Course Objective:

Understand the Mathematical representations of solids used in geometric construction.

Course Outcome:

Understand and develop the Mathematical representations of solids used in geometric construction.

NO OF LECTURE HOURS: 7

LECTURE	LECTURE TOPIC	KEY ELEMENTS	LEARNING OBJECTIVES (2 to 3 objectives)
1.	Geometry and Topology of Solids	Geometry of Solids	Understanding geometry and topology of solids (B5, B6)
2.	Comparison of wireframe, surface and solid models	Comparing wireframe, surface and solid models	Learning Comparison of wireframe, surface and solid models (B3, B5)
3.	Properties of solid model	Solid model properties	Learning about Properties of solid model (B4, B5)
4.	Properties of representation schemes	Representation schemes	Learning properties of representation schemes (B4, B5)
5.	Concept of Half-spaces, Boolean operations.	Half spaces and Boolean operations	Learning about Concept of Half-spaces, Boolean operations. (B4, B5)
6.	Schemes: B-rep, CSG, Sweep representation, ASM	Schemes	Learning about Schemes: B-rep, CSG, Sweep representation, ASM (B4, B5)
7.	Primitive instancing, Cell Decomposition and Octree encoding	Schemes	Learning about Primitive instancing, Cell Decomposition and Octree encoding (B4, B5)

Wire Frame Modeling

- A geometric model of an object is created by using the two-dimensional geometric entities such as: points, lines, curves, circles etc.
- Very often, designers build physical models to help in the visualization of a design. This may require the construction of skeleton model using wires to represent the edges of an object or component.
- Wire frame modeling is used in computer aided engineering techniques and also to facilitate the production of various projected views to aid visualization.
- The model appears like a frame constructed out of wire hence it is called as WIRE FRAME model.
- It is simple to construct.
- It requires less computer memory for storage compare to other types of geometric models.
- The time required to retrieve, edit or update is less for wire-frame models compare to other types of geometric models.
- It is more ambiguous to interpret the wire-frame model (see Fig. 3.1).

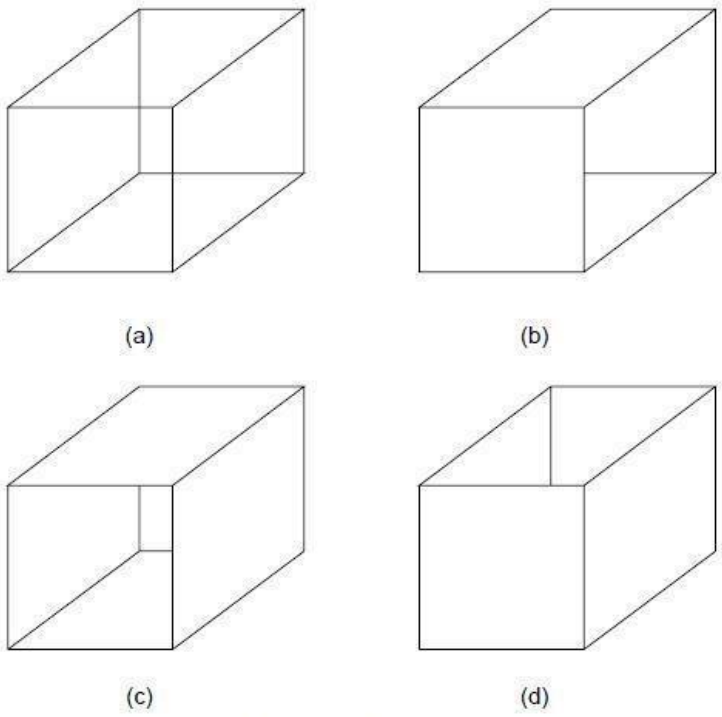


Fig. 3.1 Ambiguity in Wire frame Model

- Calculation of the properties, such as mass, volume, moment of inertia etc., is not possible.
- It is difficult and time consuming to generate wire-frame model for complicated objects.
- It requires more input data compare to others as it requires co-ordinates of each node and their connectivity.

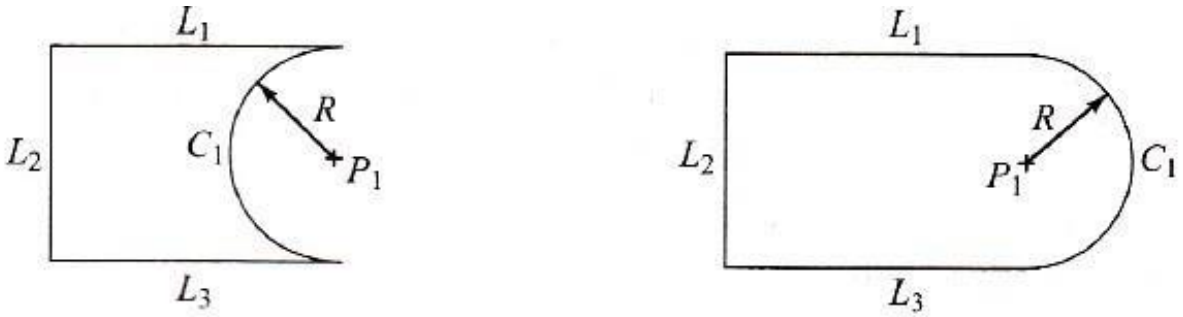
Solid Modeling

- To eliminate all kinds of ambiguities in representation and manipulations of the objects, the solid modeler was developed. Out of the various approaches developed, one of them was an approach to the design of mechanical parts by treating them as combinations of simple building blocks like cylinders and cuboids.
- Such solid modelers or volume modelers can hold complete unambiguous representations of the geometry of a wide range of solid objects. The completeness of the information contained in a solid model allows the automatic production of realistic images of a shape and automation of the process of interference checking.
- Furthermore, interfaces can interrogate the model and extract lot of useful data. The model can also serve as a means of geometric input for finite element analysis or even manufacturing tasks such as the generation of instructions for numerically controlled machining.
- Solid modelers store more information (geometry and topology) than wireframe or surface modelers (geometry only). Both wireframe and surface models are incapable of handling spatial addressability as well as verifying that the model is well formed, the latter meaning that these models cannot verify whether two objects occupy the same space.
- Surface models provide a precise definition of surfaces and can handle complex geometries, they are slow to render, are computationally intensive and do not further CAD/CAM automation and integration goals. A shaded surface model is by no means considered a solid model.
- On the other hand, solid modeling produces accurate designs, provides complete three-dimensional definition, improves the quality of design, improves visualization and has potential for functional automation and integration.
- However solid modeling has some limitations. For example, it cannot automatically create other models from the solid definition and neither can it automatically use data created in other models to create a solid. In addition, solid modeling has not been proven for large-scale production applications.
- Other limitations such as slow rendering and computations as well as poor user interface are fading away with the rapid enhancement of both hardware and software.

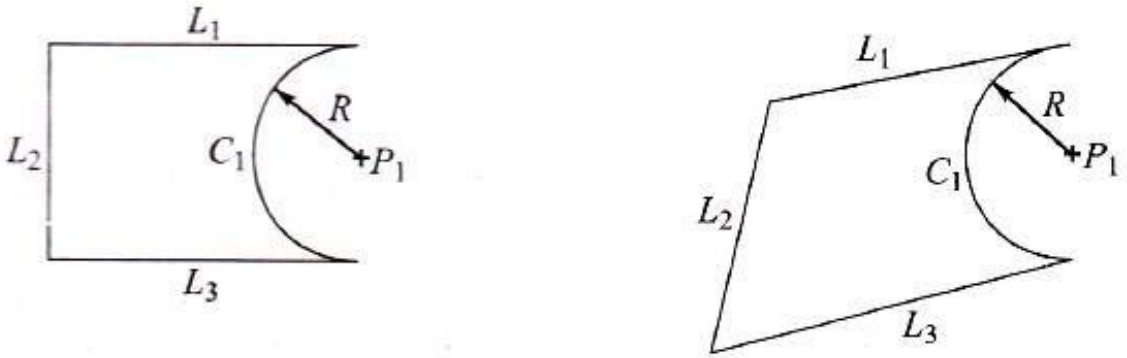
Geometry and Topology

- The difference between geometry and topology is illustrated in Fig. 3.2. Geometry (sometimes called metric information) is the actual dimensions that define the entities of the object.
- The geometry that defines the object shown in Fig. 3.2 is the lengths of lines L_1, L_2 and L_3, angles between the lines and the radius R and the center P_1 of thehalf-circle.
- Topology (sometimes called combinatorial structure), on the other hand, is the connectivity and associativity of the object entities.

- It has to do with the notion of neighborhood; that is, it determines the relational information between object entities.
- The topology of the object shown in Fig. 3.2b can be stated as follows: L_1 shares a vertex (point) with L_2 and C_1, L_2 shares a vertex with L_1 and L_3, L_3 shares a vertex with L_2 and C_1, L_1 and L_3 do not overlap and P_1 lies outside the object. Based on these definitions, neither geometry nor topology alone can completely model objects.
- Wireframe and surface models deal only with geometrical information of objects and are therefore considered incomplete and ambiguous. From a user point of view, geometry is visible and topology is considered to be nongraphical relational information that is stored in solid model databases and are not visible to users.



(*a*) Same geometry but different topology



(*b*) Same topology but different geometry

Fig. 3.2 Difference between Geometry and Topology of an Object

Properties of solid model

The properties that a solid model or an abstract solid should capture mathematically can be stated as follows:

1. **Rigidity** This implies that the shape of a solid model is invariant and does not depend on the model location or orientation in space.
2. **Homogeneous three-dimensionality** Solid boundaries must be in contact with the interior. No isolated or dangling boundaries (see Fig. 3.3) should be permitted.

3. **Finiteness and finite describability** The former property means that the size of the solid is not infinite while the latter ensures that a limited amount of information can describe the solid. The latter property is needed in order to be able to store solid models into computers whose storage space is always limited. It should be noted that the former property does not include the latter and vice versa. For example, a cylinder which may have a finite radius and length may be described by an infinite number of planar faces.
4. **Closure under rigid motion and regularized boolean operations** This property ensures that manipulation of solids by moving them in space or changing them via boolean operations must produce other valid solids.
5. **Boundary determinism** The boundary of a solid must contain the solid and hence must determine distinctively the interior of the solid.

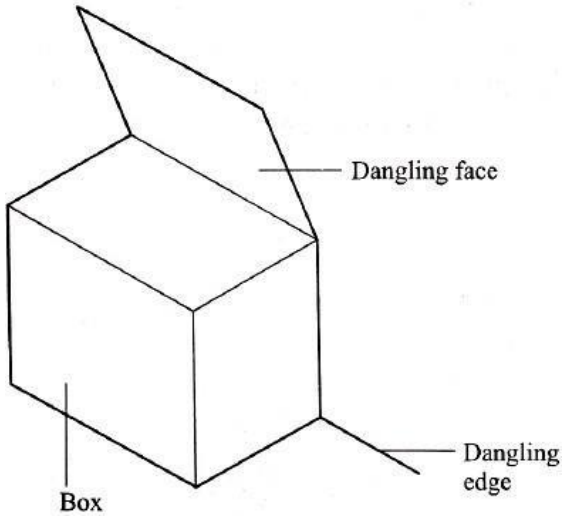


Fig. 3.3

Properties of representation scheme

The formal properties of representation schemes which determine their usefulness in geometric modeling can be stated as follows:

1. **Domain:** The domain of a representation scheme is the class of objects that the scheme can represent or it is the geometric coverage of the scheme.
2. **Validity** The validity of a representation scheme is determined by its range, that is, the set of valid representations or models it can produce. If a scheme produces an invalid model, the CAD/CAM system in use may crash or the model database may be lost or corrupted if an algorithm is invoked on the model database. Validity checks can be achieved in three ways: test the resulting databases via a given algorithm, build checks into the scheme generator itself, or design scheme elements (such as primitives) that can be manipulated via a given syntax.
3. **Completeness or Unambiguousness** This property determines the ability of the scheme to support analysis and other engineering applications. A complete scheme must provide models with sufficient data for any geometric calculation to be performed on them.

Uniqueness This property is useful to determine object equality. It is a custom in algebra to check for uniqueness but it is rare to do so in geometry. This is because it is difficult to develop algorithms to detect the equivalence of two objects and it is computationally expensive to implement these algorithms if they exist. Positional and permutational nonuniqueness are two simple cases shown in fig. 3.4. Fig. 3.4(a) shows a two dimensional rectangular solid (of side lengths a and b) in two different positions and orientations. The two-dimensional solid S shown in fig. 3.4(b) is divided into three blocks A, B and C that can be unioned in a different order.

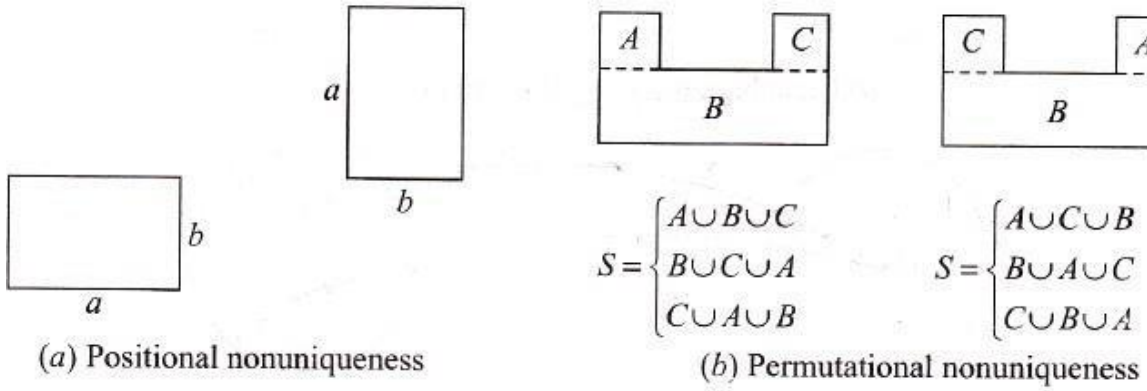


Fig. 3.4

The informal properties of representation schemes for solid models

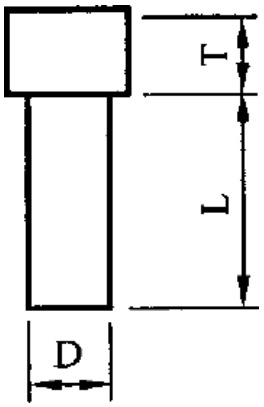
The primary considerations for storing the model in a computer are:

i. **Conciseness** (i.e. how much computer memory is occupied by the model of a shape).
ii. **Efficiency** (i.e. how much computer time is used in creating, interrogating or modifying the model of a shape.

- In general, there is a trade-off between the usage of memory and usage of time. For example, if we use up storage for 'remembering' the properties of a shape, then we do not need to spend a lot of time recalculating these properties every time one needs the information.
- Representation schemes where the elements of a shape are explicitly held in the model are termed as *evaluated representations.* Conversely, those where the elements must be calculated from implicit instructions for construction of the shape are known as *unevaluated representations.*
- Depending on the application, either of the above approaches can be used. Representation of solid in a computer can be divided into 6 general classes as follows:

1. Pure Primitive Instancing

- This technique is used to define numerous families of objects, each defined parametrically as shown in *Fig. 3.5*. A particular solid is specified completely by giving the family to which it belongs, together with a limited set of parameter values. However, such systems can only define a restricted range of objects which have been predefined in the system.

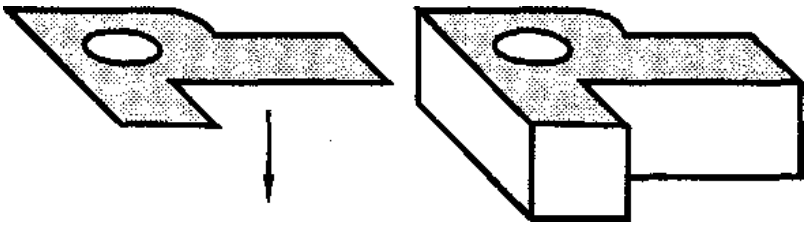


Pin parameters (D, T, and L)

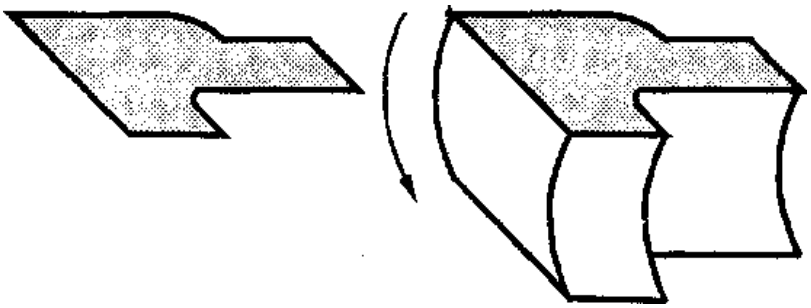
Fig. 3.5 Pure Primitive Instancing

2. Generalized Sweeps

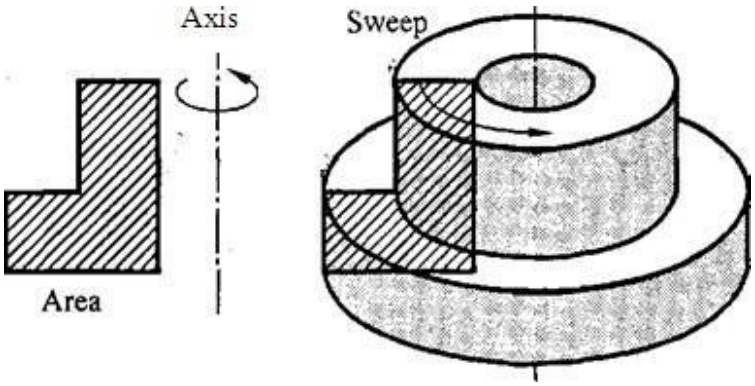
– A solid is defined in terms of volumes swept out by 2D or 3D lamina as they move along a curve. Some of the typical objects generated by this method are shown in *Fig. 3.6*.



(a) Translational Sweep



(b) Translational Sweep over Path

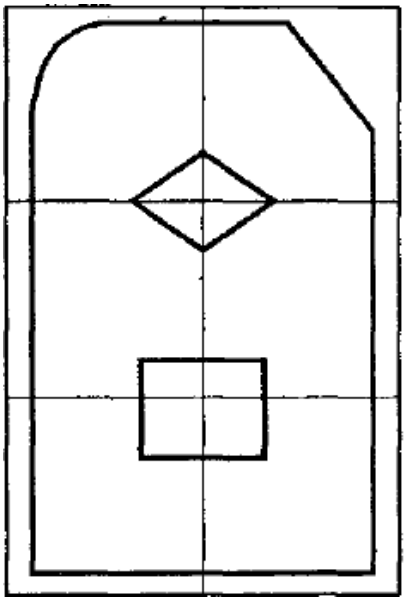


(c) Rotational Sweep

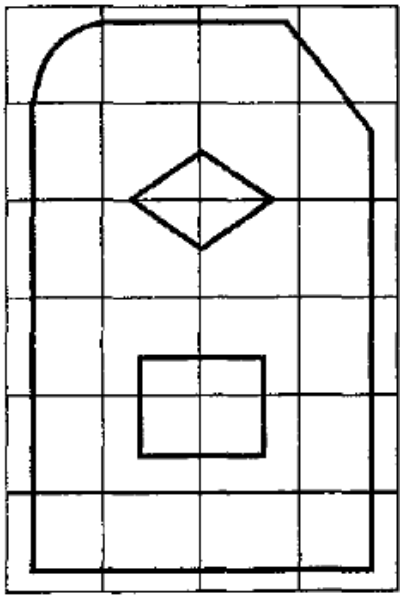
Fig. 3.6 Generalised Sweep

3. Spatial Occupancy Enumeration

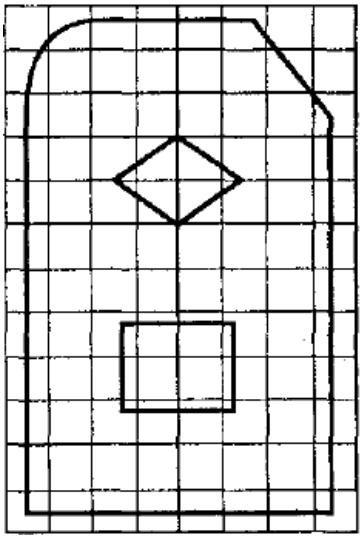
- Under this technique 3D objects are divided up into cubical cells at a particular resolution and objects are modeled by listing the cells that they occupy as shown in *Fig. 3.7* (a) to (c).
- Smaller the size of the cube, more accurate would be the representation. This representation scheme requires large amounts of storage for reasonable resolution and thus has not been favored in practical systems.



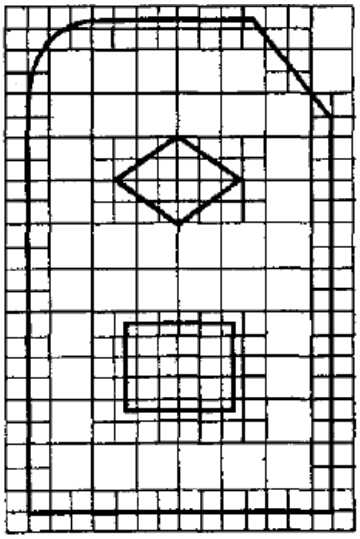
(a) First Level



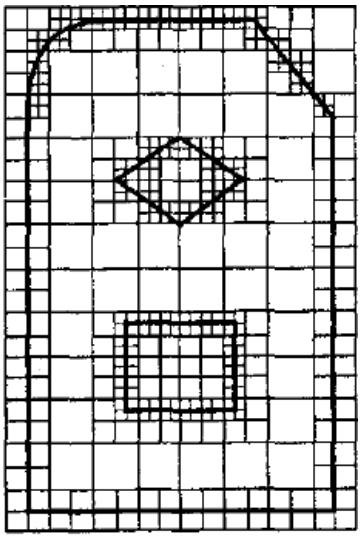
(b) Second Level



(c) Third Level



(d) Fourth Level



(e) Fifth Level

Fig. 3.7 Spatial Occupancy Enumeration and Octree Decomposition

- However, this problem has been reduced in a development of a scheme known as *Octree decomposition* [Refer *Fig. 3.7* (d) & (e)]. In this scheme only those cells that are occupied at each level are stored and sub-division is only necessary for partially occupied cells.

4. Cellular Decomposition

- An object is represented in this scheme by a list of cells it occupies, but the cells are not necessarily cubes, nor are they necessarily identical (see *Fig. 3.8*)

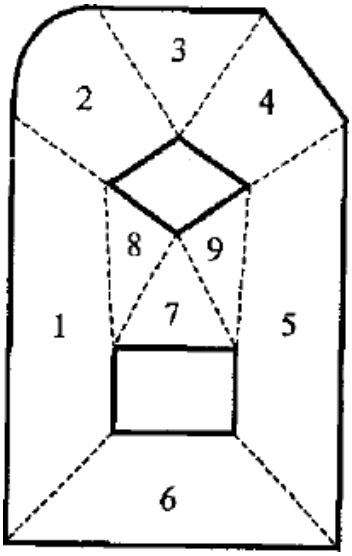


Fig. 3.8 Cellular Decomposition (9 Cell Decomposition)

5. Constructive Solid Geometry (CSG or C-rep)

- This method is also called as the *Building Block Approach.* The CSG system allows the user to build the model out of solid graphic primitives, such as rectangular blocks, spheres, cylinders, cones, wedges and torus. These shapes and the mathematical representation of these blocks are shown below in *Fig. 3.9*.

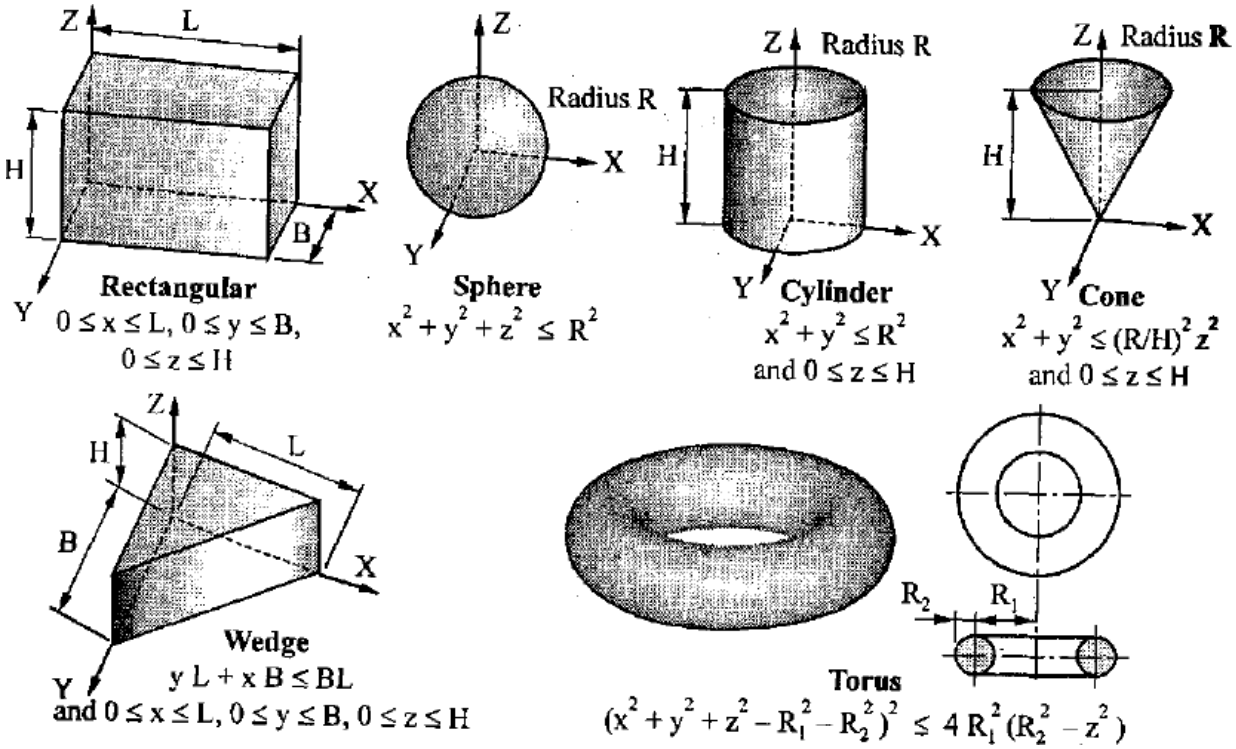


Fig. 3.9 Solid Primitives

- This is one of the most popular method of representing and building complex solids. ACSG model is based on the principle that any physical object can be divided into a set of bounded primitives which can be combined in a certain manner to form the physical object.
- Structuring and combining the primitives of the solid model in the graphics database, is achieved by the use of Boolean operations. Consider two solids A and B, sharing a common volume of space as shown in *Fig. 3.10* (a). The Boolean operations that can be performed on them and the resultant solid are shown in *Fig. 3.10* (b) to (d).

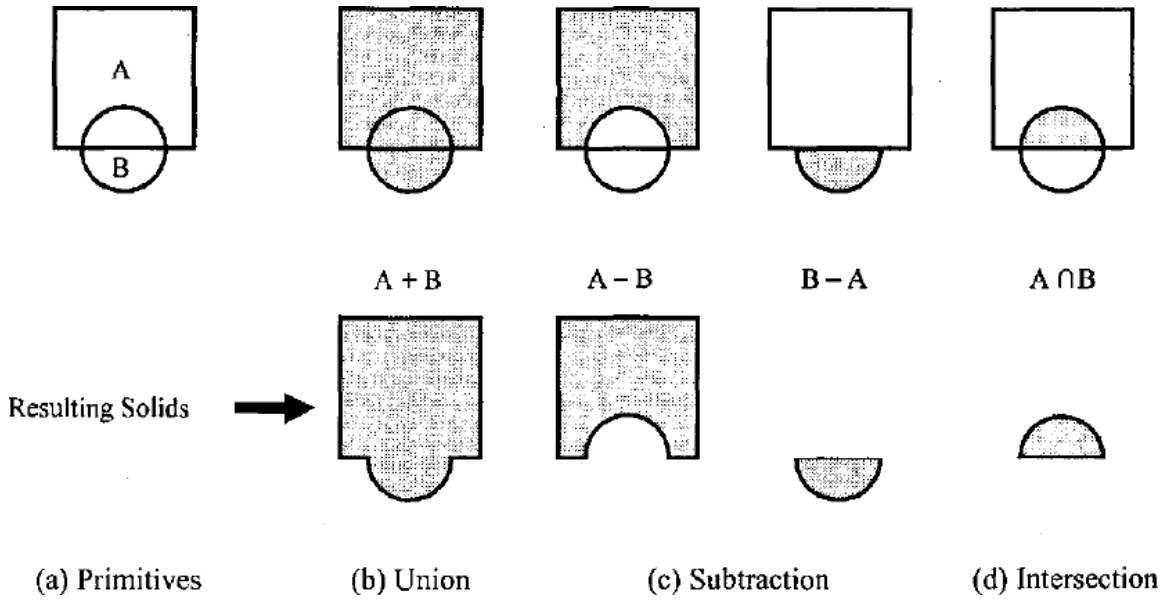
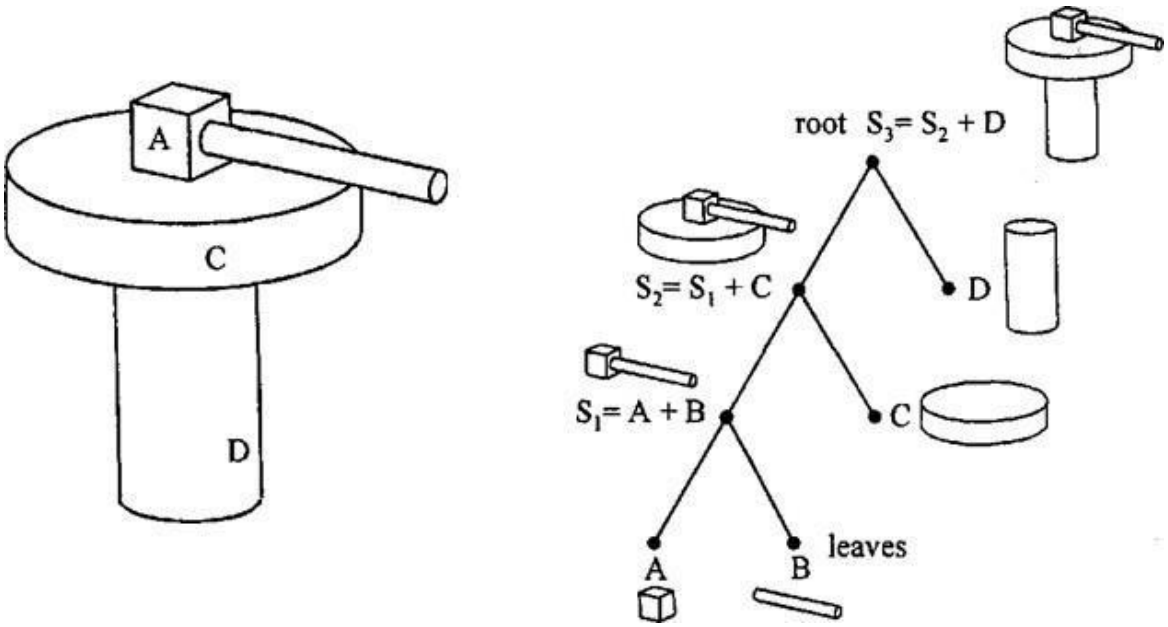
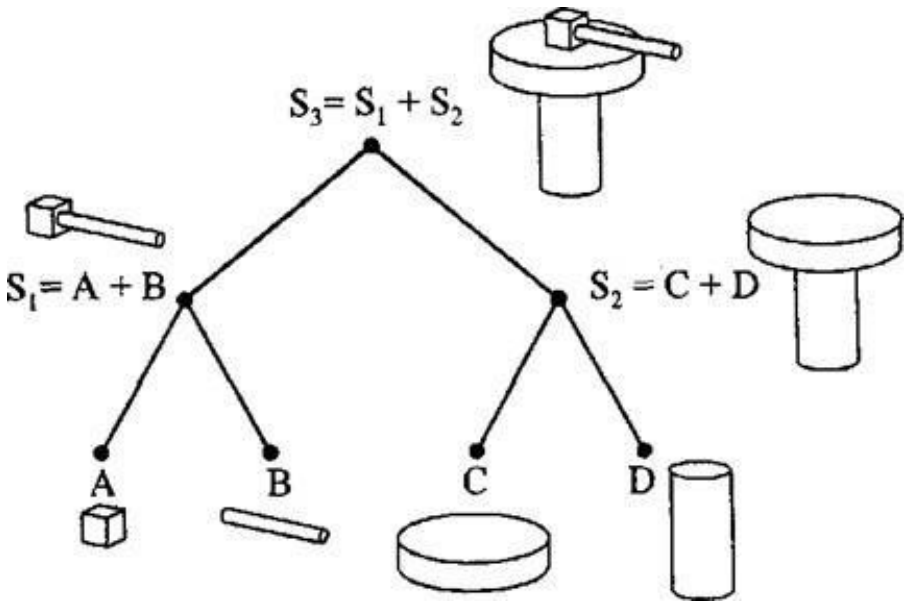


Fig. 3.10 Boolean Operations

- CSG method has a procedural advantage in the initial formulation of the model. It is relatively easy to construct a precise solid model out of regular solid primitives by adding, subtracting and intersecting the components. CSG uses the unevaluated representation format which results in compact file of the model in the database. However, this results in more computations to evaluate the model.
- The data representation of CSG objects is represented by a *binary tree.* The binary tree gives the complete information of how individual primitives are combined to represent the object. The number of primitives thus decides the number of Boolean operations required to construct the binary tree.
- The balanced distribution of the tree is desired to achieve minimum computations while modifying or interrogating a model. A balanced tree can be defined as a tree whose left and right subtrees have almost an equal number of nodes.
- The creation of balanced or an unbalanced tree is solely dependent on the user and is related to how the primitives are combined. Ideally the model should be built from an almost central position and branch out in two opposite directions.
- For example, while doing a union operation, it is preferred to combine all the unions together, rather than achieving unions of objects two at a time. A balanced and an unbalanced method of building the same object is shown in *Fig. 3.11*.



(a) Unbalanced Tree



(b) Balanced Tree

Fig. 3.11 CSG Tree

- The CSG tree is organized upside down, with the *root* representing the composite object at the top and primitives called as *leaves* at the bottom.

6. Boundary Representation (B-rep)

- The boundary representation approach or B-rep is a popular method of solid representation. It can be considered as an extension of the wire frame model with additional face information added.
- In a B-rep, the solid is considered to be bounded by its surface and has its *interior* and *exterior.* The surface consists of a set of well organized faces. The two important areas for B-rep models are the *topological* and *geometrical* information.

- Topological information provides the relationship about vertices, edges and faces similar to that used in a wire frame model. In addition to connectivity, topological information also includes *orientation* of edges and faces. Geometric information is usually in terms of equations of the edges and the faces.
- In a B-rep model, the information related to the orientation of the face is important. Generally external vertices of the face are numbered in anti-clockwise fashion. The normal vector of the face as determined by the right hand rule must point to the exterior of the surface.
- Consider the three faces as shown in *Fig. 3.12*. One of the methods to describe the top face would be 4-3-5-6. On similar lines the other two faces could be described as 1-2-3-4 and 1-4-6-3.

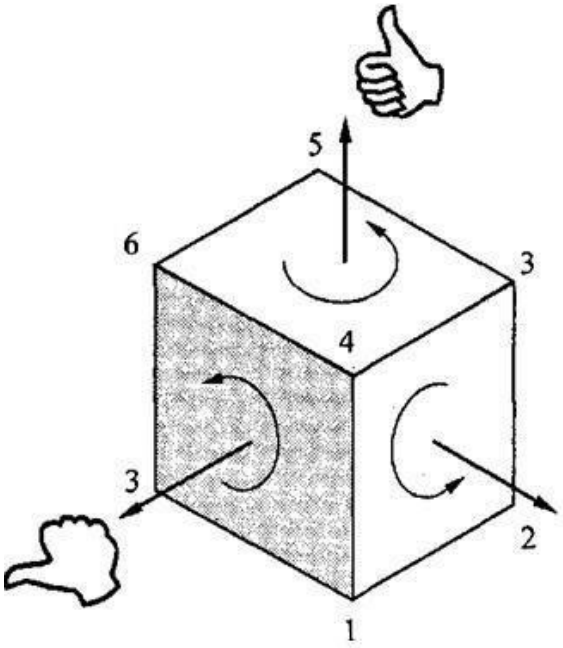
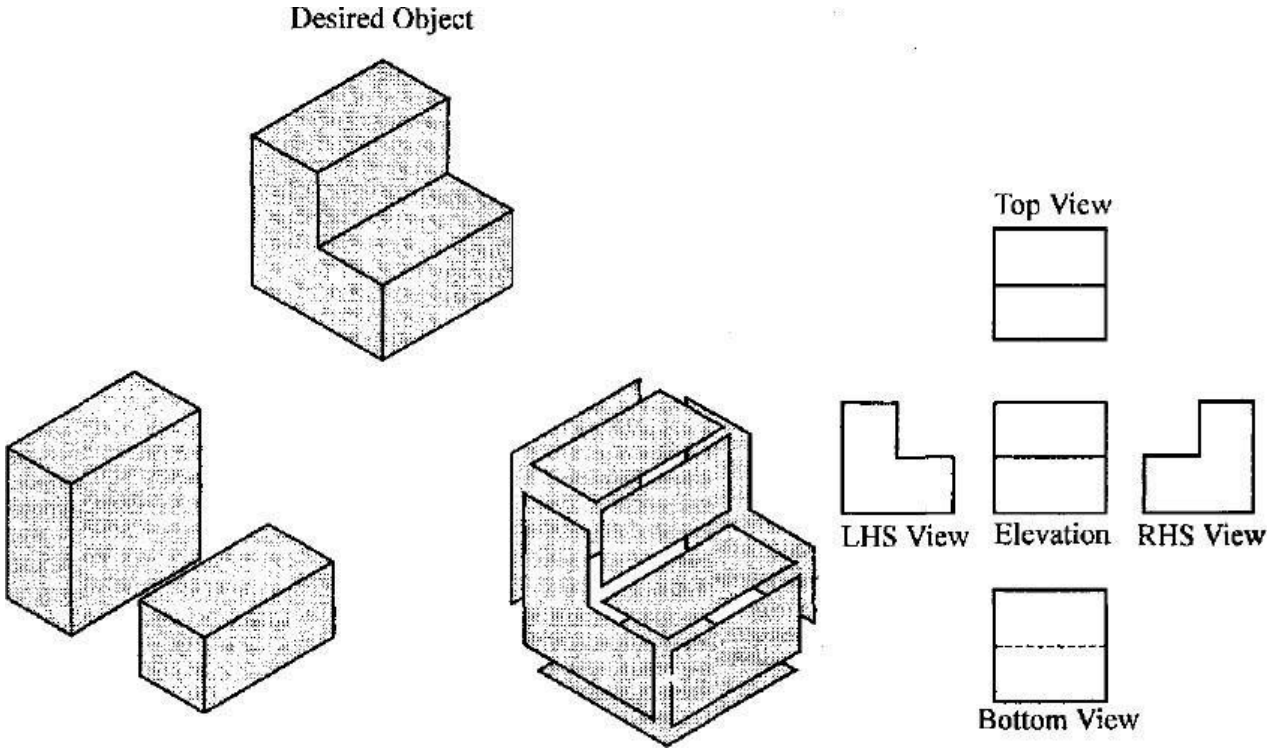


Fig. 3.12 Orientation of External Face

- However, not all surfaces can be oriented in this way. If the surface of the solid can be oriented, it is called *orientable;* otherwise it is *non-orientable.*
- The CSG and the B-rep information of the same solid are shown in *Fig. 3.13*. The CSG model is achieved by the union of the two primitives whereas the facial information for B-rep is as shown.
- It is seen that B-rep systems store only the bounding surfaces of the solid, however it is still possible to compute geometrical and mass properties by virtue of numerical methods such as Gauss Divergence Theorem, which relates volume integrals to surface ones. The speed and accuracy of computations vary as per the kind of surfaces used.
- B-rep systems are extremely useful when unusual shapes are encountered, which would not be possible with CSG. This kind of situation occurs in aircraft fuselage, wing shapes and in automobile body styling. Such shapes would be extremely difficult to build by CSG.

– In B-rep models, the solid is represented as an evaluated data structure. This requires more storage space but does not demand nearly the same computation as would be required for CSG to reconstruct the image. This makes it more efficient.



(a) CSG Information (b) B-rep Information

Fig. 3.13 Comparison of CSG and B-rep Information

– Another advantage of B-rep is that, it is relatively simple to convert back and forth between a boundary representation and a corresponding wire-frame model. The reason is that the model's boundary definition is similar to the wire-frame definition which facilitates conversion of one form to another.

Table 3.1 Comparison of C-rep and B-rep

C-rep	B-rep
It does not store only the bounding surfaces of the solid.	It stores only the bounding surfaces of the solid.
It is easy to create precise solid model using standard primitives.	It does not use standard primitives.
It is difficult to create unusual shapes.	It is easy to create unusual shapes.
Less storage space.	More storage space.
More computation time required.	Less computation time required.
It is relatively difficult to convert back and forth between a C-rep and a corresponding wire-frame model.	It is relatively simple to convert back and forth between a boundary representation and a corresponding wire-frame model.

7. Skinning or Lofting

- This method is uses a number of two dimensional profiles for generating a three dimensional object. The two dimensional profiles are arranged as desired and a three dimensional space curve is made to pass through these profiles.
- A skinning operation is then done for generating a solid. A typical example is shown in *Fig. 3.14.* This method is useful in modeling turbine blades, aerofoil shapes, manifolds, volute chambers and other such surfaces.

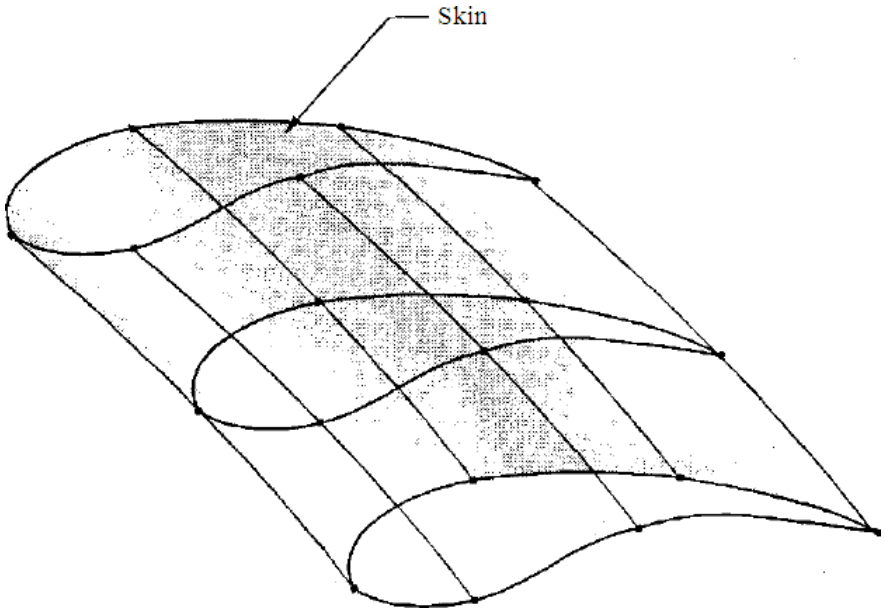


Fig. 3.14 Skinning

8. Miscellaneous Modeling Techniques:

- A complete modeler should have certain additional features which would simplify the modeling procedures. A few of them are described below:

(a) Tweaking :

- A model can be created by using a combination of methods as described previously. The ability available to the user to modify or move a vertex or a face and to see the effect of that on the model is referred to as tweaking. This feature is useful in styling automobile bodies and other consumer products.

(b) Filleting :

- Most engineering and consumer components are provided with smooth edges or generous fillets. This not only improves the stress distribution, but also prevents injury while handling them.
- If filleting procedure is done through CSG method, it would require a lot of data input for achieving it. Instead direct procedures which allow the user to input the properties of a fillet are necessary. This would result in a quick filleting operation, without a large database.
- A typical filleting operation in wire frame and rendered modes are shown in *Fig. 3.15*.

- Each of the solid modeling representation discussed above has its advantages and disadvantages. The CSG and B-rep approaches are widely used in major CAD softwares.
- In a *single representation modeler*, the representation is usually B-rep, with certain commands such as sweep and a few CSG like inputs are accepted. However, the storing of data is in B-rep format only.
- As compared to this, a *dual representation modeler* has a primary representation scheme as CSG. (e. g. PADL-2). Here the modeler has both B-rep and CSG representations. However, in this case, B-rep is derived internally and the user has no control over it.
- In a truly hybrid *modeler* there are two independent internal representations of CSG and B-rep. With these systems, the users have the capability to construct the geometric model by either approach (CSG or B-rep), whichever is more appropriate to the particular problem.

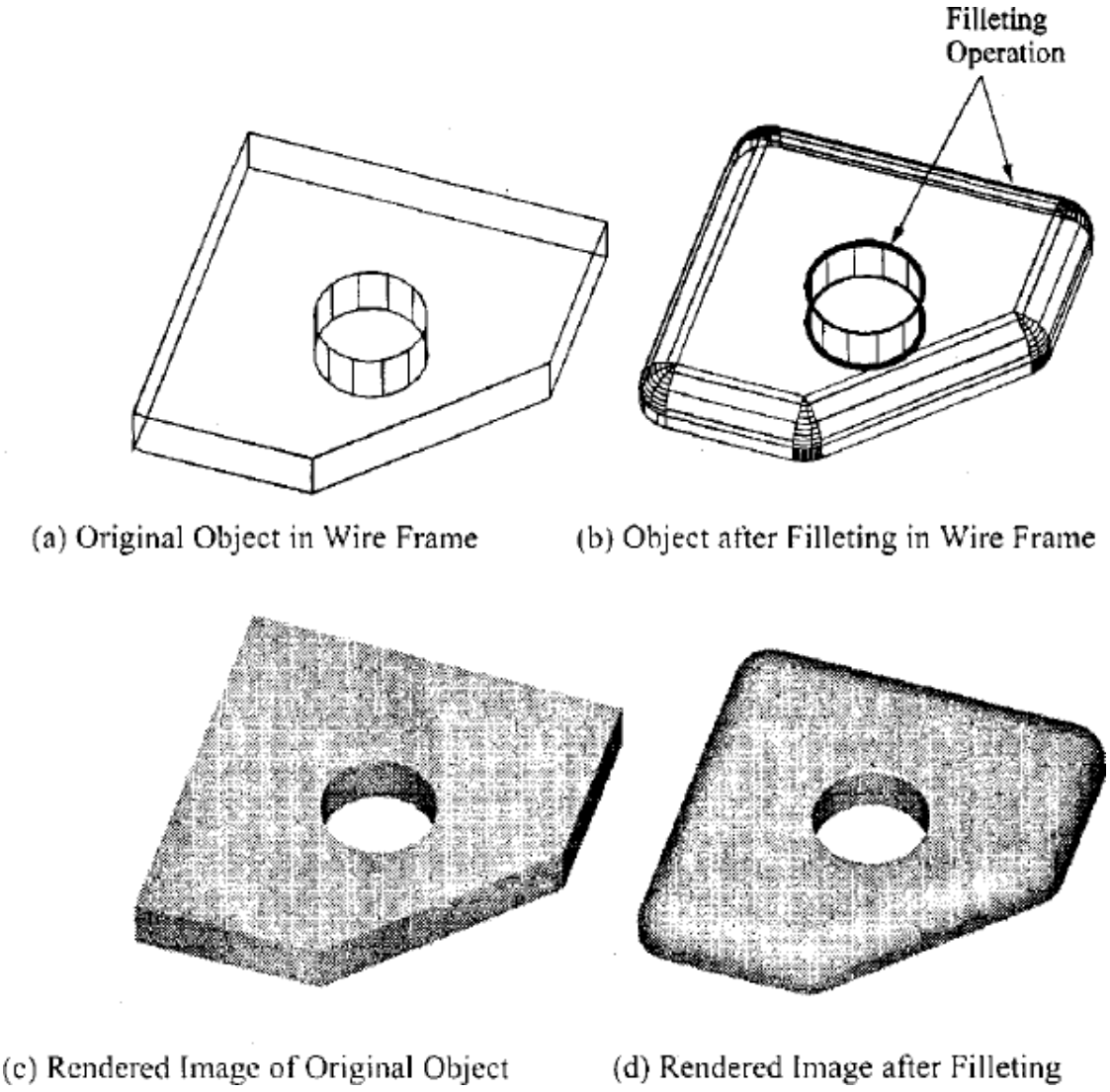


Fig. 3.15 Filleting Operation

Half spaces

- Half-spaces form a basic representation scheme for bounded solids. By combining half-spaces (using set operations) in a building block fashion, various solids can be constructed. Half-spaces are usually unbounded geometric entities; each one of them divides the representation space into two infinite portions, one filled with material and the other empty. Surfaces can be considered half-space boundaries and half-spaces can be considered directed surfaces.
- A half-space is defined as a regular point set in three dimensional coordinate system (E^3) as follows:

  $H = \{P: P \in E^3 \text{ and } f(P) < 0\}$
- where P is a point in E^3 and f(P) = 0 defines the surface equation of the half-space boundaries. Half-spaces can be combined together using set operations to create complex objects.
- The most used half spaces are:
  Planar half-space: $H=\{(x, y, z): z < 0\}$
  Cylindrical half-space: $H=\{x, y, z): x^2 + y^2 < R^2\}$
  Spherical half-space: $H = \{(x, y, z): x^2 + y^2 + z^2 < R^2\}$
  Conical half-space: $H = \{(x, y, z): x^2 + y^2 < [(\tan \alpha/2)z]^2]\}$
  Toroidal half-space: $H = \{(x, y, z): (x^2 + y^2 + z^2 – R_2^2 – R_1^2)^2 < 4 R_2^2(R_1^2– z^2)\}$

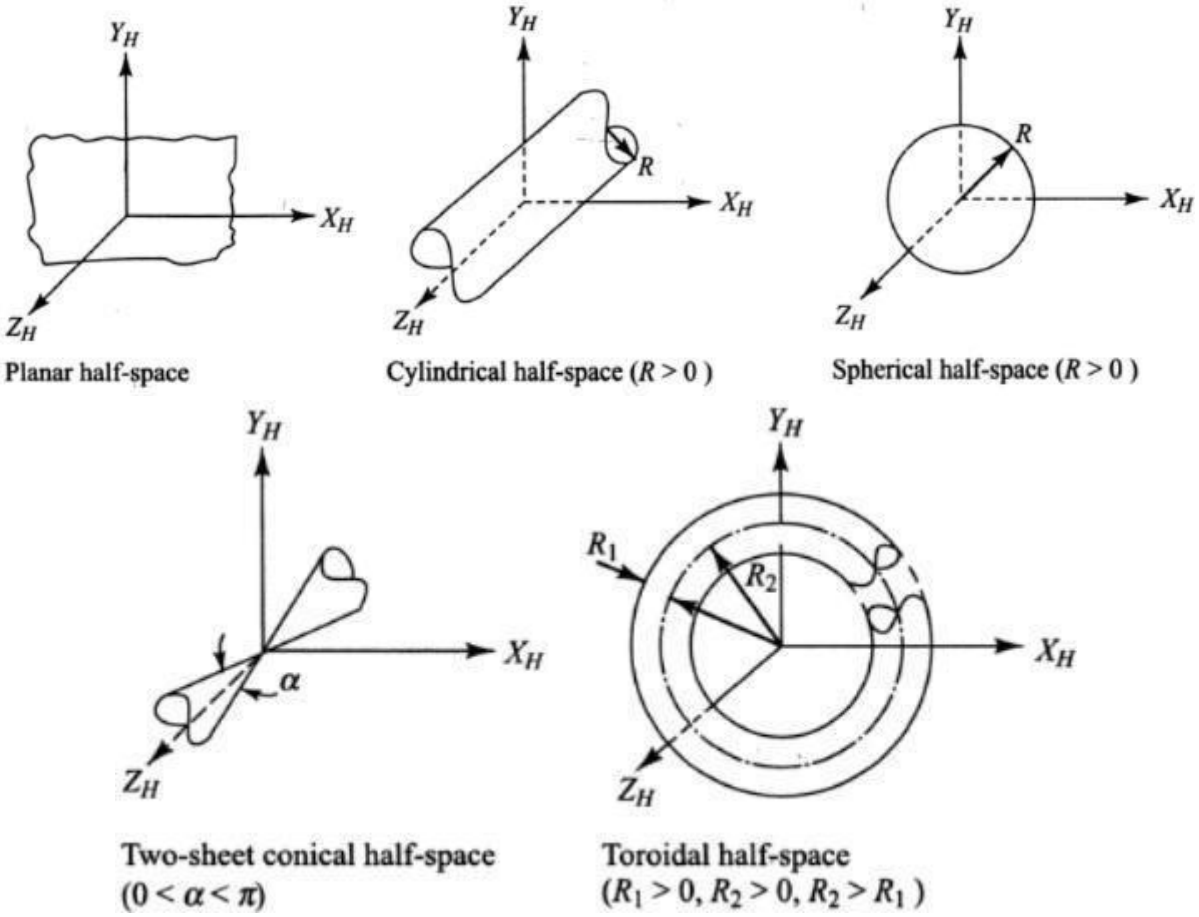


Fig. 3.16

Tutorial Questions

1. Explain about Wireframe modeling.
2. Explain about geometry and topology of solids.
3. Explain properties of solids.
4. Explain properties of representation schemes.
5. Explain concept of Half spaces.
6. Explain about different Boolean operations.
7. Explain about B-representations.

QUESTION BANK

1. Explain about Wireframe modeling.
2. Explain about geometry, topology and properties of solids.
3. Explain properties of representation schemes.
4. Explain concept of Half spaces.
5. Explain about B-representations.

UNIT 4

GEOMETRIC TRANSFORMATIONS

Course Objective:

Understand the transformation of 2D and 3D parts.

Course Outcome:

Able to get the transformed in 2D and 3D using transformation equations.

NO OF LECTURE HOURS: 8

LECTURE	LECTURE TOPIC	KEY ELEMENTS	LEARNING OBJECTIVES (2 to 3 objectives)
1.	Geometric Transformations: Homogeneous representation	Geometric Transformations	Understanding homogeneous representation of Geometric Transformations (B5, B6)
2.	Translation, Scaling,	Translation, Scaling, Reflection	Learning about Translation, Scaling, Reflection (B4, B5)
3.	Reflection	Reflection	Learning about Reflection (B4, B5)
4.	Rotation, Shearing in 2D and 3D	Rotation, Shearing in 2D and 3D	Learning about Rotation, Shearing in 2D and 3D (B4, B5)
5.	Orthographic projections	Orthographic projections	Learning about Orthographic projections (B4, B5)
6.	Perspective projections	Perspective projections	Learning about Perspective projections (B4, B5)
7.	Window to View-port transformation.	Window to View-port transformation.	Learning about Window to View-port transformation. (B4, B5)
8.	Examples	Examples	Solving different examples (B2, B3)

Introduction

- Computer graphics can be defined as creation, storage and manipulation of pictures and drawings by means of a computer.
- It enhances the power of communication between the computer and its users.
- It allows representation of huge set of numbers in the form of a graph or a picture which helps in better understanding of the characteristic and pattern of data.
- A viewing surface of a raster scan display is divided into an array of pixels. A CRT screen is made up of horizontal and vertical lines.
- Each horizontal line is made up of pixels. The memory of the computer is a collection of bits and a bit can take any value either 0 or 1.
- Each pixel can be represented by one bit in memory. This memory is called frame buffer.
- If a line as to be represented, then the pixel must be made bright to display the line.
- The pixel value of bit must be 1 in memory. This is known as scan conversion.

Geometric Transformations

- All changes performed on the graphic image are done by changing the database of the original picture. These changes are called as transformations.
- Transformations allow the user to uniformly change the entire picture. An object created by the user is stored in the form of a database. If the database, which represents the object, is changed, the object also would change. This method is used to alter the orientation, scale, position of the drawing.
- In general, geometric transformations can be defined as the change done to the database by performing certain mathematical operations on it, so as to produce the desired change in the image.

4.2.1 Translation

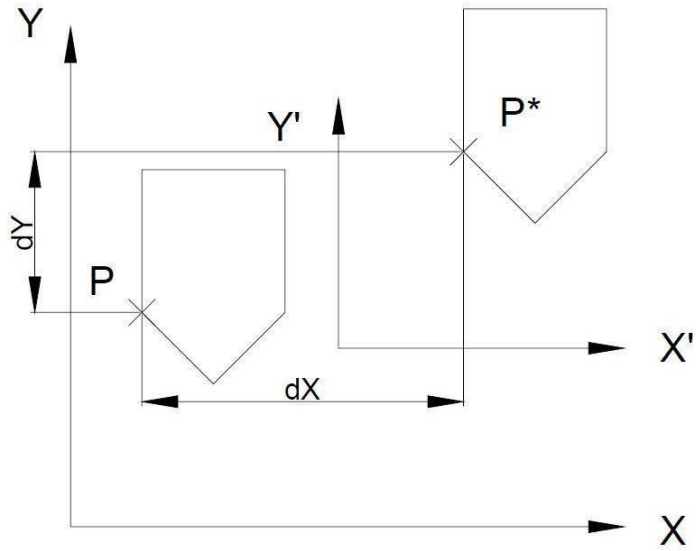


Fig. 4.1 Translation

- When every entity of a geometric model remains parallel to its initial position, the transformation is called as translation.
- Translating a model therefore implies that every point on it moves by an equal given distance in a given direction. A translation involves moving of an element from one location to another.

$$P^* = [x^*, y^*]$$

$$x^* = x + dx$$

$$y^* = y + dy$$

$$[P^*] = \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x+dx \\ y+dy \end{bmatrix}$$

$$= \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

Scaling

- Scaling transformation alters the size of an object. Scaling can be uniform (i.e. equal in both X and Y directions) or non-uniform (i.e. different in X and Y directions).
- To achieve scaling, the original coordinates would be multiplied by scaling factor.

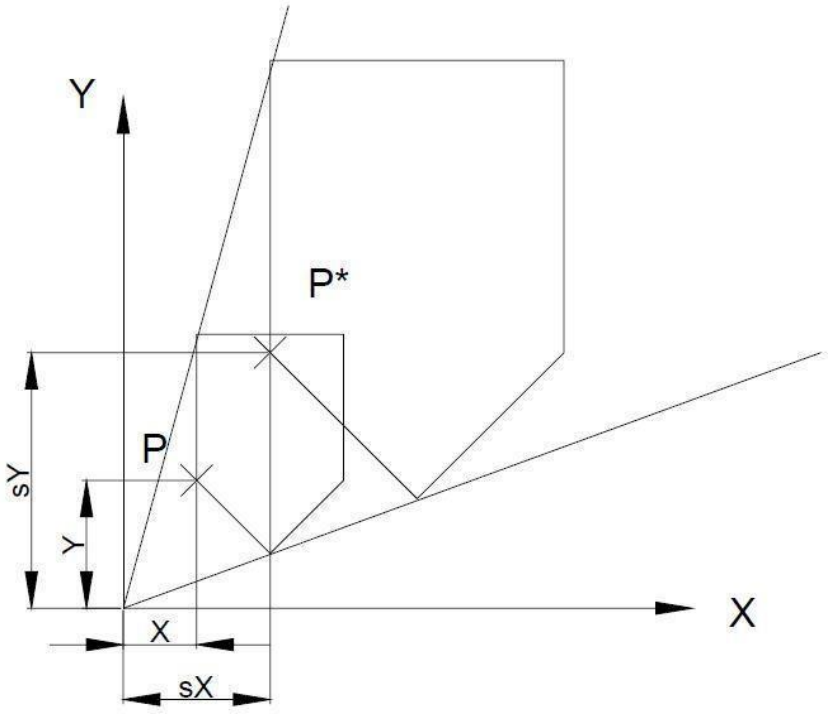


Fig. 4.2 Scaling

$$P^* = [x^*, y^*] = [S_x \times x, S_y \times y]$$

This equation can also be represented in matrix form as follows:

$$[P^*] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Reflection or Mirror

- Reflection is the process of obtaining a mirror of the original shape.
- This is an important transformation and is used quite often as many engineered products are symmetrical.
- The following transformation matrices as shown in Fig, 4.3 when multiplied to the original point produce a Reflection or Mirror.
- Thus, in general,

$$P' = P \cdot M$$

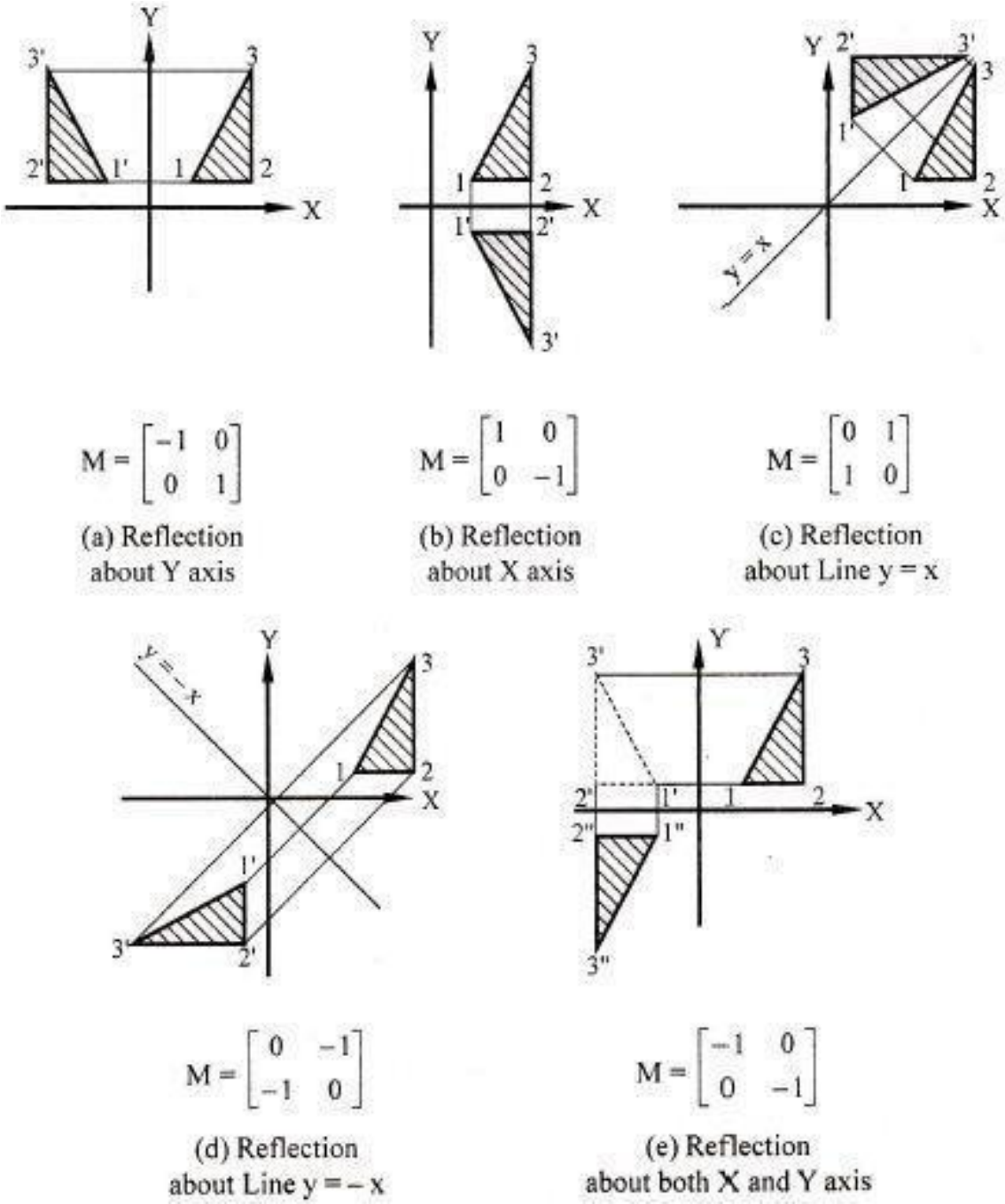


Fig. 4.3 Reflection

Rotation

- Rotation is an important transformation and allows the user to view the objects from different angles.
- It can also be used to create entities arranged in circular arrays, by creating the entity once and then rotating / copying it to the desired positions on the circumference. This would allow the user to create an array of objects.

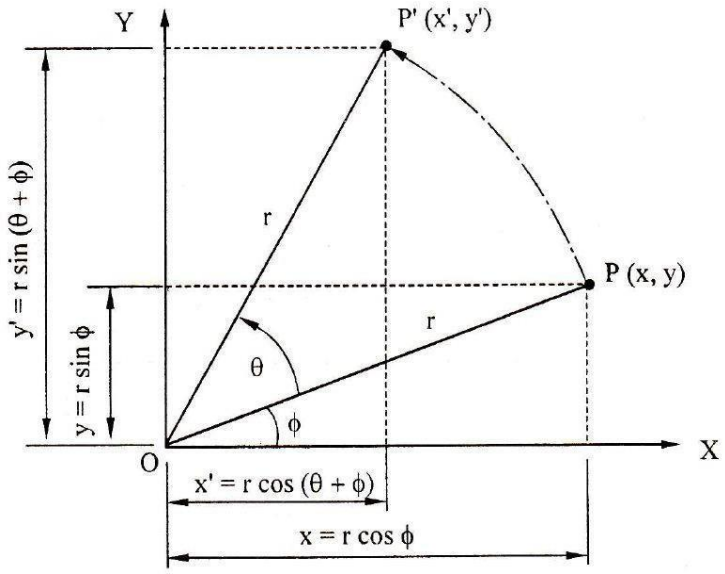


Fig. 4.4 Rotation

- Let point P (x, y) be rotated by an angle θ about the origin O.

$$P = \begin{bmatrix} x & y \end{bmatrix} = \begin{bmatrix} r\,\mathrm{Cos}\,\phi & r\,\mathrm{Sin}\,\phi \end{bmatrix}$$

- Let the rotated point be represented as :

$$P' = \begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} r\,\mathrm{Cos}\,(\theta+\phi) & r\,\mathrm{Sin}\,(\theta+\phi) \end{bmatrix}$$

$$P' = \begin{bmatrix} r(\mathrm{Cos}\theta\,\mathrm{Cos}\phi - \mathrm{Sin}\theta\,\mathrm{Sin}\phi) & r(\mathrm{Sin}\theta\,\mathrm{Cos}\phi + \mathrm{Cos}\theta\,\mathrm{Sin}\phi) \end{bmatrix}$$

$$\therefore P' = \begin{bmatrix} (x\,\mathrm{Cos}\theta - y\,\mathrm{Sin}\theta)(x\,\mathrm{Sin}\theta + y\,\mathrm{Cos}\theta) \end{bmatrix}$$

- This can be expressed as:

$$[P'] = [x\ y]\begin{bmatrix} Cos\theta & Sin\theta \\ -Sin\theta & Cos\theta \end{bmatrix}$$

Homogeneous Representation

- Many a times it becomes necessary to combine the above mentioned individual transformations in order to achieve the required results. In such cases the combined transformation matrix can be obtained by multiplying the respective transformation matrices.
- However, care should to be taken that the order of the matrix multiplication be done in the same way as that of the transformations as follows.

$$[P^*] = [T_n]\ [T_{n-1}]\ [T_{n-2}+ \ldots\ldots.*T_3]\ [T_2]\ [T_1] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (4.1)$$

  In Eq. (4.1) all the transformation matrices should be multiplicative type.
- The following form should be used to convert the translation into a multiplication form.

$$[P^*] = [x^* \quad y^* \quad 1]$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dX & dY & 1 \end{bmatrix} [x \quad y \quad 1]$$

- Hence the translation matrix in multiplication form can be given as

$$[MT] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dX & dY & 1 \end{bmatrix}$$

- This is termed as homogeneous representation. In homogeneous representation, an n-dimensional space is mapped into (n+1) dimensional space. Thus a 2 dimensional point [x y] is represented by 3 dimensions as [x y 1]. Homogeneous representation can be experienced in the following situations.

Rotation about an arbitrary Point

- The transformation given earlier for rotation is about the origin of the axes system. It may sometimes be necessary to get the rotation about any arbitrary base point as shown in Fig. 4.5. To derive the necessary transformation matrix, the following complex procedure comprising the followin1g three points would be required.

(i) Translate the point A to O, the origin of the axes system.
(ii) Rotate the object by the given angle.
(iii) Translate the point back to its original position.

- The transformation matrices for the above operations in the given sequence are the following.

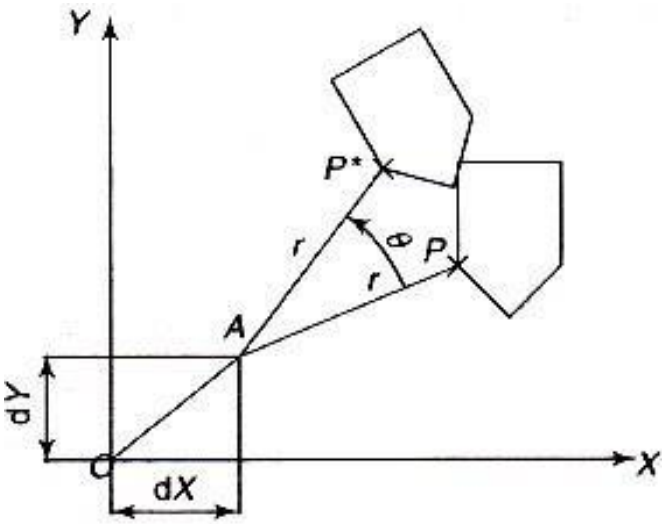


Fig. 4.5 Rotation about an arbitrary Point

$$[T_1]=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ dX & dY & 1\end{bmatrix}$$

$$[T_2]=\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$[T_3]=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -dX & -dY & 1\end{bmatrix}$$

The required transformation matrix is given by

$$[T] = [T_3]\,[T_2]\,[T_1]$$

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

Reflection about an arbitrary Line

- Similar to the above, there are times when the reflection is to be taken about an arbitrary line as shown in Fig. 4.6.
  1. Translate the mirror line along Y axis such that line passes through the origin, O.
  2. Rotate the mirror line such that it coincides with the X axis.
  3. Mirror the object through the X axis.
  4. Rotate the mirror line back to the original angle with X axis.

5. Translate the mirror line along the Y axis back to the original position.

Following are the transformation matrices for the above operations in the givensequence.

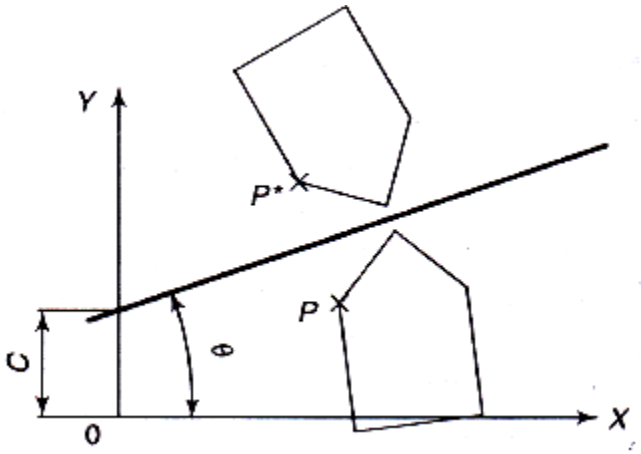


Fig. 4.6 Reflection Transformation about on arbitrary Line

$$[T_1]=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & C & 1\end{bmatrix}$$

$$[T_2]=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$[T_3]=\begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$[T_4]=\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$[T_5]=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -C & 1\end{bmatrix}$$

- The required transformation matrix is given by

$$[T] = [T_5]\,[T_4]\,[T_3]\,[T_2]\,[T_1]$$

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

Shearing

- A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called a shear.

2D Shearing

- Two common shearing transformations are those that shift coordinate x values and those that shift y values.
- An x-direction shear relative to the x axis is produced with the transformation matrix

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{.....(4.2)}$$

- which transforms coordinate positions as

$$x' = x + sh_x \cdot y, \qquad y' = y \qquad \text{.....(4.3)}$$

- Any real number can be assigned to the shear parameter sh_x. A coordinate position (x, y) is then shifted horizontally by an amount proportional to its distance (y value) from the x axis (y = 0).
- Setting sh_x to 2, for example, changes the square in Fig. 4.7 into a parallelogram. Negative values for sh_x shift coordinate positions to the left.

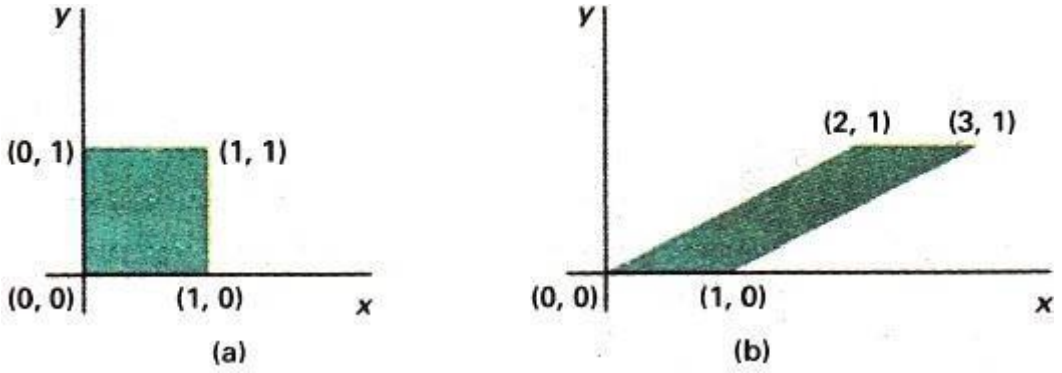


Fig. 4.7 A unit square (a) is converted to a parallelogram (b) using the x direction shear matrix 4.2 with $sh_x = 2$.

- We can generate x-direction shears relative to other reference lines with

$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{.....(4.4)}$$

- with coordinate positions transformed as

  $x' = x + sh_x (y - y_{ref})$, $y' = y$ ..... (4.5)

- An example of this shearing transformation is given in Fig.4.8 for a shear parameter value of 1 /2 relative to the line $y_{ref} = -1$.

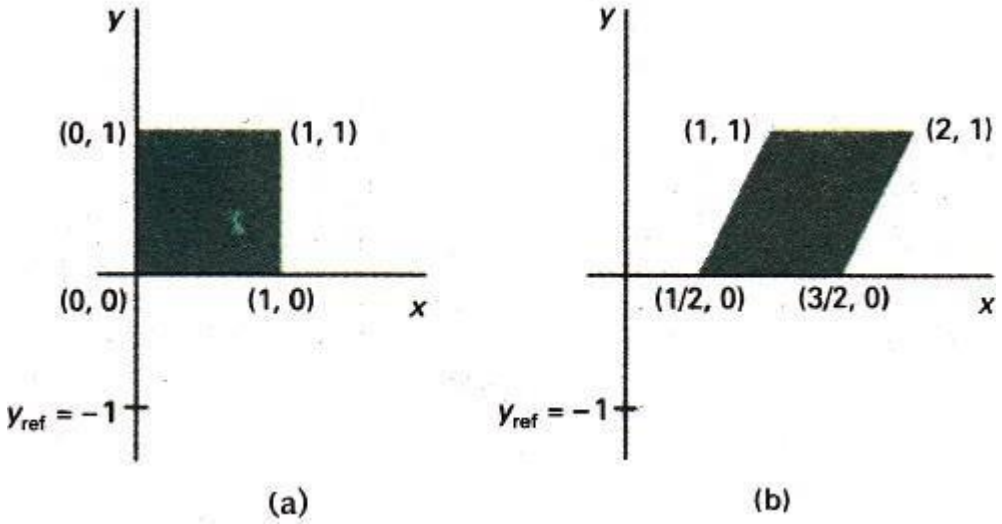


Fig. 4.8 A unit square (a) is transformed to a shifted parallelogram (b) with $sh_x = 1/2$ and $y_{ref} = -1$ in the shear matrix 4.4.

- A y-direction shear relative to line $x = x_{ref}$ is generated with transformation matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & -sh_y \cdot x_{ref} \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{.....(4.6)}$$

- which generates transformed coordinate positions

  $x' = x$, $y' = y + sh_y (x - x_{ref})$ ..... (4.7)

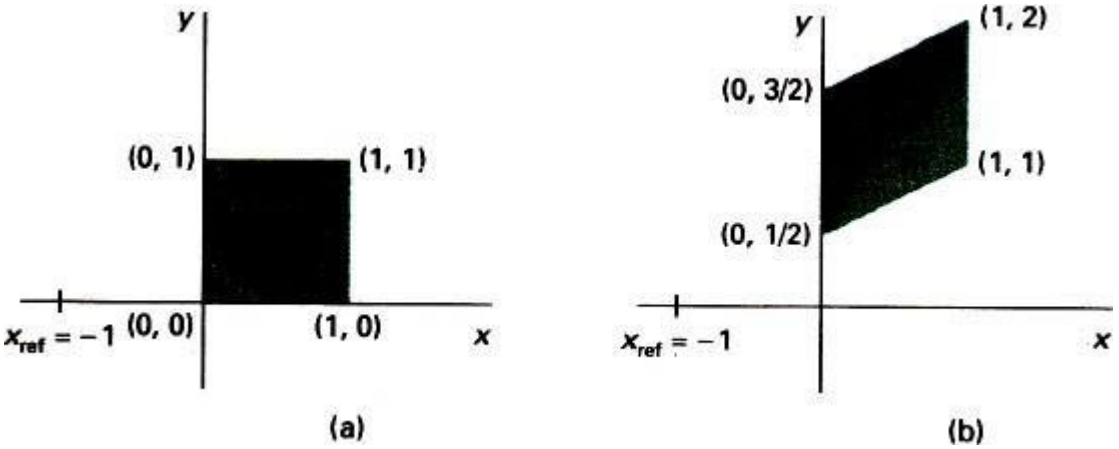


Fig. 4.9 A unit square (a) is turned into a shifted parallelogram (b) with parameter values $sh_y = 1/2$ and $x_{ref} = -1$ in the y direction using shearing transformation 4.6.

- This transformation shifts a coordinate position vertically by an amount proportional to its distance from the reference line $x = x_{ref}$. Fig. 4.9 illustrates the conversion of a square into a parallelogram with $sh_y = 1/2$ and $x_{ref} = -1$.
- Shearing operations can be expressed as sequences of basic transformations. The x-direction shear matrix 4.2, for example, can be written as a composite transformation involving a series of rotation and scaling matrices that would scale the unit square of Fig. 4.9 along its diagonal, while maintaining the original lengths and orientations of edges parallel to the x axis.
- Shifts in the positions of objects relative to shearing reference lines are equivalent to translations.

3D Shearing

- In two dimensions, transformations relative to the x or y axes to produce distortions in the shapes of objects. In three dimensions, we can also generate shears relative to the z axis.

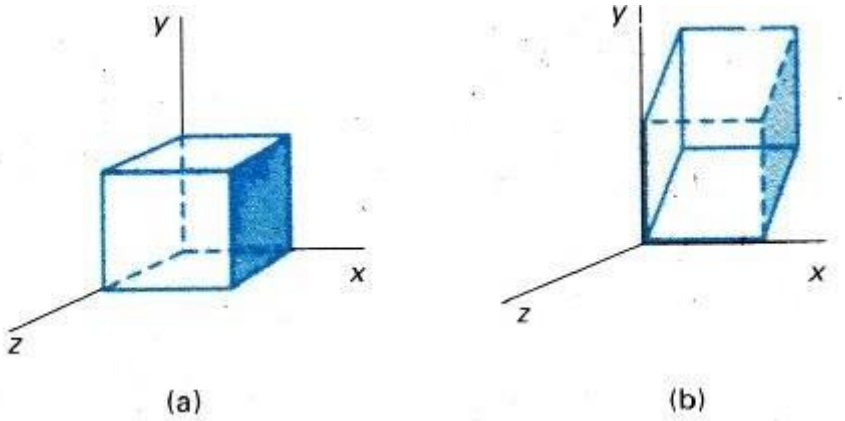


Fig 4.10

- As an example of three-dimensional shearing, the following transformation produces a z-axis shear:

$$[SH_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Parameters a and b can be assigned any real values. The effect of this transformation matrix is to alter x- and y-coordinate values by an amount that is proportional to the z value, while leaving the z coordinate unchanged.
- Boundaries of planes that are perpendicular to the z axis are thus shifted by an amount proportional to z. An example of the effect of this shearing matrix on a unit cube is shown in Fig. 4.10, for shearing values a=b=1. Shearing matrices for the x axis and y axis are defined similarly.

4.2 Projections of Geometric Models

- Databases of geometric models can only be viewed and examined if they can be displayed in various views on a display device or screen. Viewing a three dimensional model is a rather complex process due to the fact that display devices can only display graphics on two-dimensional screens.
- This mismatch between three-dimensional models and two-dimensional screens can be resolved by utilizing projections that transform three-dimensional models onto a two-dimensional projection plane. Various views of a model can be generated using various projection planes.
- To define a projection, a center of projection and a projection plane must be defined as shown in Fig. 4.11. To obtain the projection of an entity (a line connecting points P_1 and P_2 in the figure), projection rays (called projectors) are constructed by connecting the center of projection with each point of the entity.

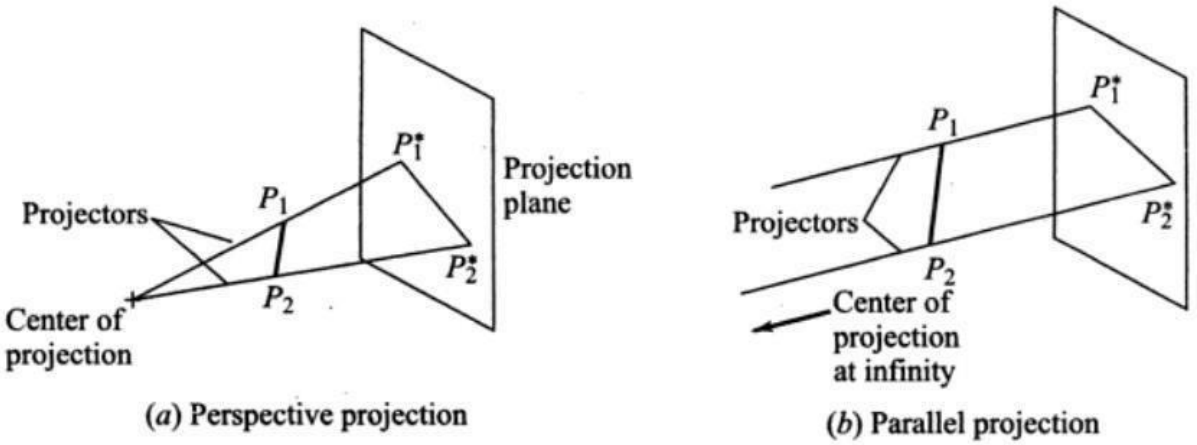


Fig. 4.11 Projection Definition

- The intersections of these projectors with the projection plane define the projected points which are connected to produce the projected entity. There are two different types of projections based on the location of the center of projection relative to the

projection plane.

- If the center is at a finite distance from the plane, perspective projection results and all the projectors meet at the center. If, on the other hand, the center is at an infinite distance, all the projectors become parallel (meet at infinity) and parallel projection results. Perspective projection is usually a part of perspective, or projective, geometry.
- Such geometry does not preserve parallelism, that is, no two lines are parallel. Parallel projection is a part of affine geometry which is identical to Euclidean geometry. In affine geometry, parallelism is an important concept and therefore is preserved.

Perspective projection

- Perspective projection creates an artistic effect that adds some realism to perspective views. As can be seen from Fig. 4.11(a), the size of an entity is inversely proportional to its distance from the center of projection; that is, the closer the entity to the center, the larger its size is.
- Perspective views are not popular among engineers and draftsmen because actual dimensions and angles of objects and therefore shapes, cannot be preserved, which implies that measurements cannot be taken from perspective views directly. In addition, perspective projection does not preserve parallelism.
- In order to define perspective projection, a centre of projection and a projection plane are required. The centre of projection is placed along the z-axis and the image is projected onto z = 0 or xy plane.
- The transformation of perspective projection is given by:

$$P^* = Po\ P$$

where

$$P_O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{bmatrix}$$

P is the given point.

P* is the reflected point.

d is the distance between the centre of projection to the projection plane.

Orthographic Projection

- Unlike perspective projection, parallel projection preserves actual dimensions and shapes of objects. It also preserves parallelism. Angles are preserved only on faces of the object which are parallel to the projection plane.
- There are two types of parallel projections based on the relation between the direction of projection and the projection plane. If this direction is normal to the projection plane, orthographic projection and views result.

- The orthographic projection system is to include a total of six projecting planes in any direction required for complete description.
- A typical example is shown in Fig. 4.12 where the object is enclosed in a box such that there are 6 mutually perpendicular projecting planes on which all possible 6 views of the object can be projected.
- This would help in obtaining all the details of the object as shown in Fig. 4.13. The visible lines are shown with the help of continuous lines while those that are not visible are shown by means of broken lines.

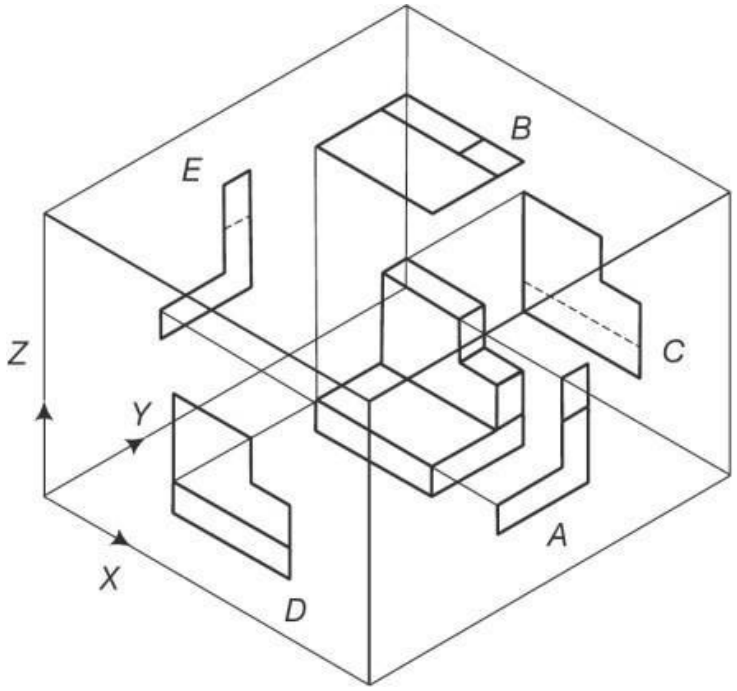


Fig. 4.12 An object Enclosed in a Cube for Obtaining Various Parallel Projections

- Obtaining the orthographic projections is relatively straight forward because of the parallel projections involved. The top view can be obtained by setting z = 0. The transformation matrix will then be

$$[M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- For obtaining the front view, y = 0 and then the resulting coordinates (x, z) will be rotated by 90^0 such that the Z axis coincides with the Y axis. The transformation matrix will then be

$$[M_{FRONT}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

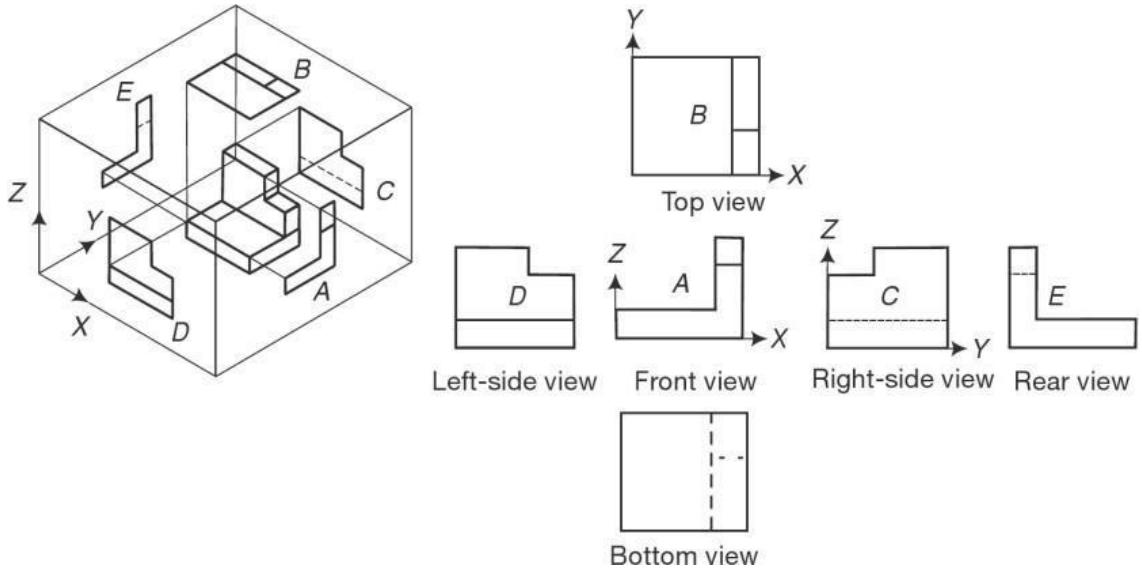


Fig. 4.13 Orthographic Projection of an Object.

- Similarly for obtaining the right side view, x = 0 and then the coordinate system is rotated such that Y axis coincides with the X axis and the Z axis coincides with the Y axis. The transformation matrix will then be

$$[M_{RIGHT}] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Window to View-port transformation

- A world-coordinate area selected for display is called a window. An area on a display device to which a window is mapped is called a viewport. The window defines what is to be viewed; the viewport defines where it is to be displayed.
- Often, windows and viewports are rectangles in standard position, with the rectangle edges parallel to the coordinate axes. Other window or viewport geometries, such as general polygon shapes and circles, are used in some applications, but these shapes take longer to process.

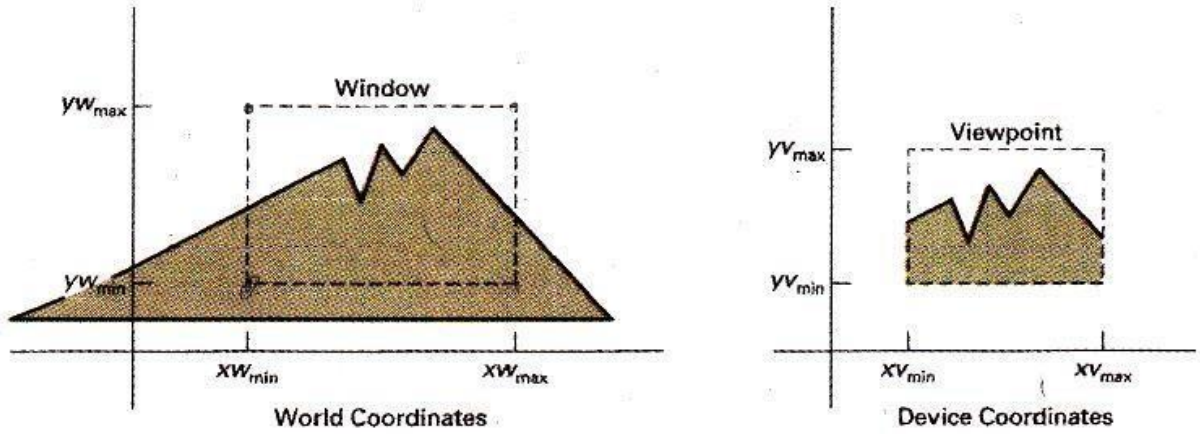


Fig. 4.14 A viewing transformation using standard rectangles for the window and viewport

- In general, the mapping of a part of a world-coordinate scene to device coordinates is referred to as a viewing transformation. Sometimes the two-dimensional viewing transformation is simply referred to as the window-to-viewport transformation or the windowing transformation.
- Fig. 4.14 illustrates the mapping of a picture section that falls within a rectangular window onto a designated rectangular viewport.
- Fig. 4.15 illustrates the window-to-viewport mapping. A point at position (xw, yw) in the window is mapped into position (xv,yv) in the associated viewport. To maintain the same relative placement in the viewport as in the window, we require that

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$
$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}} \qquad \text{.....(4.8)}$$

- Solving these expressions for the viewport position (xv, yv), we have

$$xv = xv_{min} + (xw - xw_{min})sx$$
$$yv = yv_{min} + (yw - yw_{min})sy \qquad \text{.....(4.9)}$$

- Where the scaling factors are

$$sx = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$
$$sy = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}} \qquad \text{.....(4.10)}$$

- Eq. 4.9 can also be derived with a set of transformations that converts the window area into the viewport area. This conversion is performed with the following sequence of transformations:

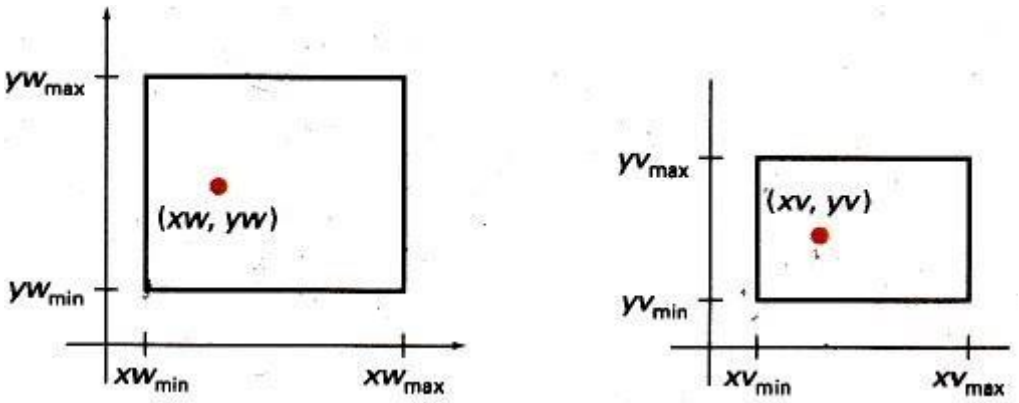


Fig. 4.15 A point at position (x_w, y_w) in a designated window is mapped to viewport coordinates (x_v,y_v) so that relative positions in the two areas are the same.

1. Perform a scaling transformation using a fixed-point position of (xw_{min}, yw_{min}) that scales the window area to the size of the viewport.
2. Translate the scaled window area to the position of the viewport.

- Relative proportions of objects are maintained if the scaling factors are the same (sx = sy). Otherwise, world objects will be stretched or contracted in either the x or y direction when displayed on the output device.
- Character strings can be handled in two ways when they are mapped to a viewport. The simplest mapping maintains a constant character size, even though the viewport area may be enlarged or reduced relative to the window.
- This method would be employed when text is formed with standard character fonts that cannot be changed. In systems that allow for changes in character size, string definitions can be windowed the same as other primitives. For characters formed with line segments, the mapping to the viewport can be carried out as a sequence of line transformations.
- From normalized coordinates, object descriptions are mapped to the various display devices. Any number of output devices can be open in a particular application, and another window-to-viewport transformation can be performed for each open output device.
- This mapping, called the workstation transformation, is accomplished by selecting a window area in normalized space and a viewport area in the coordinates of the display device.
- With the workstation transformation, we gain some additional control over the positioning of a scene on individual output devices. As illustrated in Fig. 4.16, we can use workstation transformations to partition a view so that different parts of normalized space can be displayed on different output devices.

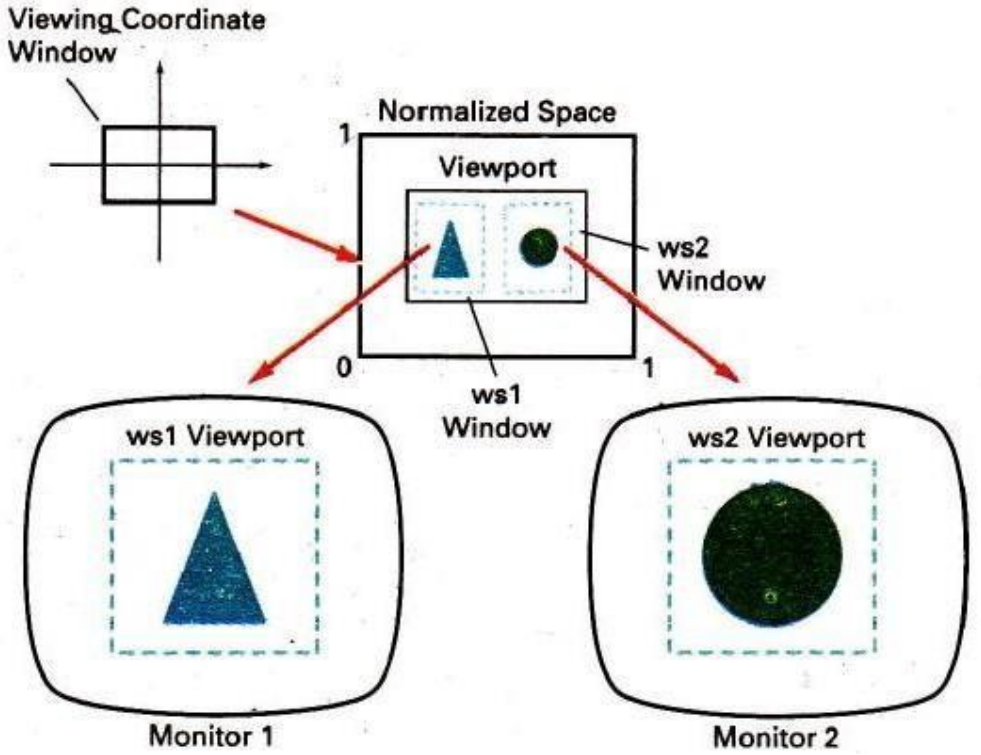


Fig. 4.16 Mapping selected parts of a scene in normalized coordinates to different video monitors with workstation transformations.

Example 4.1

A triangle ABC with vertices A (32, 22), B (88, 20) and C (32, 82) is to be scaled by a factor of 0.6 about a point X (50, 42). Determine the co-ordinates of the vertices for the scaled triangle.

Solution:

Data given:

$S_x = S_y = 0.6$

Scaling about a point X (50, 42)

So, first translate the object to origin then scale the object and then translate back the scaled object to the given point X (50, 42).

$$\therefore \Delta A'B'C' = \Delta ABC \cdot T \cdot S \cdot T^{-1}$$

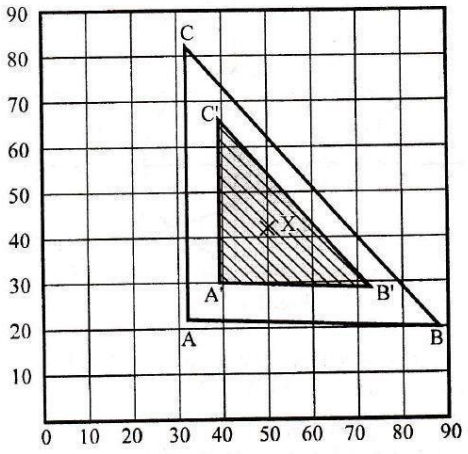
$$\therefore \Delta A'B'C' = \begin{bmatrix} 32 & 22 & 1 \\ 88 & 20 & 1 \\ 32 & 82 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -50 & -42 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 50 & 42 & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 32 & 22 & 1 \\ 88 & 20 & 1 \\ 32 & 82 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -50 & -42 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 0.6 & 0 \\ 50 & 42 & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 32 & 22 & 1 \\ 88 & 20 & 1 \\ 32 & 82 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 0.6 & 0 \\ 20 & 16.8 & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 39.2 & 30 & 1 \\ 72.8 & 28.8 & 1 \\ 39.2 & 66 & 1 \end{bmatrix}$$

Answer: New vertices are A' (39.2, 30), B' (72.8, 28.8) and C' (39.2, 66).



Example 4.2

Using transformation matrix, determine the new co-ordinates of triangle ABC with vertices A (0, 0), B (3, 2) and C (2, 3) after it is rotated by 45° clockwise about origin.

Solution:

Data given:

Rotation about origin 45° clockwise

Hence, θ = -45°

$$\therefore \Delta A'B'C' = \Delta ABC \cdot R$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 2 & 1 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} Cos(-45^{\circ}) & Sin(-45^{\circ}) & 0 \\ -Sin(-45^{\circ}) & Cos(45^{\circ}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 0 & 0 & 1 \\ 3 & 2 & 1 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 0 & 0 & 0 \\ 3.536 & -0.707 & 0 \\ 3.536 & 0.707 & 1 \end{bmatrix}$$

Answer: New vertices are A' (0, 0), B' (3.536, -0.707) and C' (3.536, 0.707).

Example 4.3

A rectangle ABCD has vertices A (1, 1), B (2, 1), C (2, 3) and D (1, 3). It has to be rotated by 30° counter clockwise about a point P (3, 2). Determine the new co-ordinates of the vertices for the rectangle.

Solution:

Data given:

Rotation about a point 45° counter clockwise

Hence, θ = 45°

Rotation about a point P (3, 2), so, first translate the object to origin then rotate the object and then translate back the rotated object to the given point P (3, 2).

$$\therefore A'B'C'D' = ABCD \cdot T \cdot R \cdot T^{-1}$$

$$\therefore A'B'C'D' = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & -2 & 1 \end{bmatrix} \begin{bmatrix} Cos30^\circ & Sin30^\circ & 0 \\ -Sin30^\circ & Cos30^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

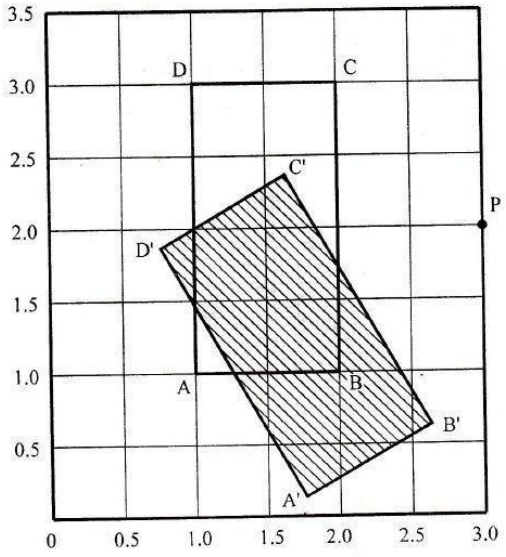
$$\therefore A'B'C'D' = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

$$\therefore A'B'C'D' = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

$$\therefore A'B'C'D' = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 1.402 & -1.232 & 1 \end{bmatrix}$$

$$\therefore A'B'C'D' = \begin{bmatrix} 1.768 & 0.134 & 1 \\ 2.634 & 0.634 & 1 \\ 1.634 & 2.366 & 1 \\ 0.768 & 1.866 & 1 \end{bmatrix}$$

Answer: New vertices are A' (1.768, 0.134), B' (2.634, 0.634), C' (1.634, 2.366) and D' (0.768, 1.866).



Example 4.4

A triangle ABC with vertices at A (0, 0), B (4, 0) and C (2, 3) is given. Perform the following operations for it.

(i) Translation through 4 and 2 units along X and Y direction respectively,

(ii) Rotation through 90° in counter clockwise direction about the new position of point C.

Solution:

(i) Translation by $t_x = 4$ and $t_y = 2$.

$$\therefore \Delta A'B'C' = \Delta ABC \cdot T$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 0 & 0 & 1 \\ 4 & 0 & 1 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 0 & 0 & 1 \\ 4 & 0 & 1 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}$$

$$\therefore \Delta A'B'C' = \begin{bmatrix} 4 & 2 & 1 \\ 8 & 2 & 1 \\ 6 & 5 & 1 \end{bmatrix}$$

Answer: New vertices are A' (4, 2), B' (8, 2) and C' (6, 5).

(ii) Rotation about the new point C' (6, 5) through 90° counter-clockwise.

Hence, θ = 90°

Rotation about a point C' (6, 5), so, first translate the object to origin then rotate the object and then translate back the rotated object to the given point C' (6, 5).

$$\therefore \Delta A''B''C'' = \Delta A'B'C' \cdot T \cdot R \cdot T^{-1}$$

$$\therefore \Delta A''B''C'' = \begin{bmatrix} 4 & 2 & 1 \\ 8 & 2 & 1 \\ 6 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -6 & -5 & 1 \end{bmatrix} \begin{bmatrix} Cos90^\circ & Sin90^\circ & 0 \\ -Sin90^\circ & Cos90^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 6 & 5 & 1 \end{bmatrix}$$

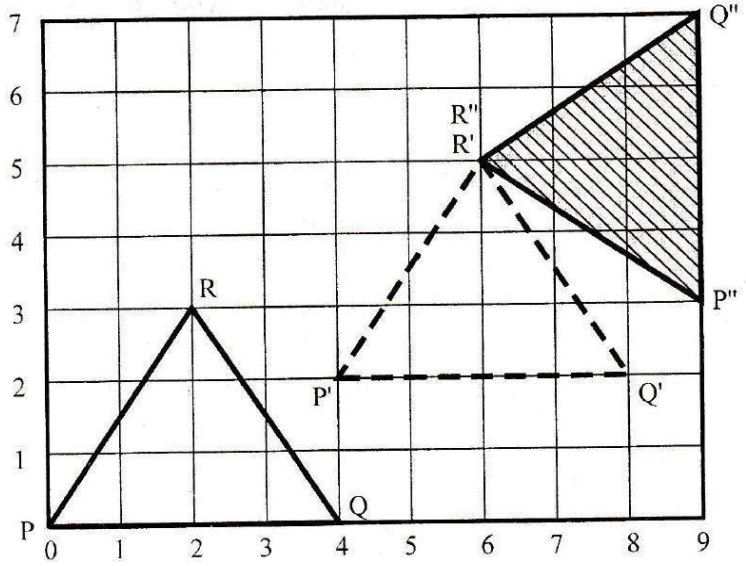
$$\therefore \Delta A''B''C'' = \begin{bmatrix} 4 & 2 & 1 \\ 8 & 2 & 1 \\ 6 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -6 & -5 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 6 & 5 & 1 \end{bmatrix}$$

$$\therefore \Delta A''B''C'' = \begin{bmatrix} 4 & 2 & 1 \\ 8 & 2 & 1 \\ 6 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -6 & -5 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 6 & 5 & 1 \end{bmatrix}$$

$$\therefore \Delta A''B''C'' = \begin{bmatrix} 4 & 2 & 1 \\ 8 & 2 & 1 \\ 6 & 5 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 11 & -1 & 1 \end{bmatrix}$$

$$\therefore \Delta A''B''C'' = \begin{bmatrix} 9 & 3 & 1 \\ 9 & 7 & 1 \\ 6 & 5 & 1 \end{bmatrix}$$

Answer: New vertices are A'' (9, 3), B'' (9, 7) and C'' (6, 5).



Example 4.5

The co-ordinates of three vertices of the triangle are P (50, 20), Q (110, 20) and R (80, 60). Determine the co-ordinates of the vertices for the new reflected triangle if it is to be reflected about:

(i) x-axis,

(ii) Line y = x.

Solution:

(i) Reflection about x-axis

$$\therefore \Delta P'Q'R' = \Delta PQR \cdot M_x$$

$$\therefore \Delta P'Q'R' = \begin{bmatrix} 50 & 20 & 1 \\ 110 & 20 & 1 \\ 80 & 60 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \Delta P'Q'R' = \begin{bmatrix} 50 & -20 & 1 \\ 110 & -20 & 1 \\ 80 & -60 & 1 \end{bmatrix}$$

Answer: New vertices are P' (50, -20), Q' (110, -20) and R' (80, -60).

(ii) Reflection about line y = x

$$\therefore \Delta P'Q'R' = \Delta PQR \cdot M_{(y=x)}$$

$$\therefore \Delta P'Q'R' = \begin{bmatrix} 50 & 20 & 1 \\ 110 & 20 & 1 \\ 80 & 60 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \Delta P'Q'R' = \begin{bmatrix} 20 & 50 & 1 \\ 20 & 110 & 1 \\ 60 & 80 & 1 \end{bmatrix}$$

Answer: New vertices are P' (20, 50), Q' (20, 110) and R' (60, 80).

Example 4.6

Reflect the diamond shape polygon whose vertices are A (-2, 0), B (0, -1) C (2, 0) and D (0, 1) about an arbitrary line L which is represented by equation y = 0.5 x + 1.

Solution:

Data given:

Reflect the polygon about line y = 0.5 x + 1

So, first translate the object to origin, and then rotate the object such that the line coincides with x-axis, then reflect the polygon about x-axis, then rotate back and then translate back.

y = 0.5 x + 1

$\therefore m = 0.5 = \tan\theta$

$\therefore \theta = \tan^{-1} 0.5 = 26.565^\circ$ (Clockwise)

$\therefore \theta = -26.565^\circ$

Also, when x = 0, y = 1 and when y = 0, x = -2.

$$\therefore t_x = 2, t_y = 0$$

$$\therefore A'B'C'D' = ABCD \cdot T \cdot R \cdot M \cdot R^{-1} \cdot T^{-1}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \begin{bmatrix} Cos\theta & Sin\theta & 0 \\ -Sin\theta & Cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cos(-\theta) & Sin(-\theta) & 0 \\ -Sin(-\theta) & Cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -t_x & -t_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cos(-26.565^\circ) & Sin(-26.565^\circ) & 0 \\ -Sin(-26.565^\circ) & Cos(-26.565^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cos26.565^\circ & Sin26.565^\circ & 0 \\ -Sin26.565^\circ & Cos26.565^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & -0.4472 & 0 \\ 0.4472 & 0.8944 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & 0.4472 & 0 \\ -0.4472 & 0.8944 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & -0.4472 & 0 \\ 0.4472 & 0.8944 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & 0.4472 & 0 \\ -0.4472 & 0.8944 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & -0.4472 & 0 \\ 0.4472 & 0.8944 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8944 & 0.4472 & 0 \\ 0.4472 & -0.8944 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0.8 & 0 \\ 0.8 & -0.6 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0.8 & 0 \\ 0.8 & -0.6 & 0 \\ -0.8 & 1.6 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 & 1 \\ -1.6 & 2.2 & 1 \\ 0.4 & 3.2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Answer: New vertices are A' (-2, 0), B' (-1.6, 2.2), C' (0.4, 3.2) and D' (0, 1).

Tutorial Questions

1. Explain about homogeneous representation of geometric transformations.
2. Explain about translation, scaling, rotation and reflecion.
3. Explain about shearing in 2D and 3D.
4. Explain about orthographic and perspective projections.
5. Explain about window to viewport transformations.

QUESTION BANK

1. A triangle ABC with vertices A (32, 22), B (88, 20) and C (32, 82) is to be scaled by a factor of 0.6 about a point X (50, 42). Determine the co-ordinates of the vertices for the scaled triangle.
2. Using transformation matrix, determine the new co-ordinates of triangle ABC with vertices A (0, 0), B (3, 2) and C (2, 3) after it is rotated by 45° clockwise about origin.
3. A triangle ABC with vertices at A (0, 0), B (4, 0) and C (2, 3) is given. Perform the following operations for it.
   (a) Translation through 4 and 2 units along X and Y direction respectively,
   (b) Rotation through 90° in counter clockwise direction about the new position of point C.
4. Explain about orthographic and perspective projections.
5. Explain about window to viewport transformations.

UNIT 5

VISUAL REALISM

Course Objective:

Understand the algorithm for visualization of various 2D and 3D parts.

Course Outcome:

Understand and develop the algorithm for visualization of various 2D and 3D parts.

NO OF LECTURE HOURS: 8

LECTURE	LECTURE TOPIC	KEY ELEMENTS	LEARNING OBJECTIVES (2 to 3 objectives)
1.	Introduction to Visual Realism	Visual Realism	Learning about Visual Realism (B4, B5)
2.	Hidden Line Removal Algorithm	Hidden Line Removal Algorithm	Learning about Hidden Line Removal Algorithm (B4, B5)
3.	Hidden surface Removal Algorithm	Hidden surface Removal Algorithm	Learning about Hidden surface Removal Algorithm (B4, B5)
4.	Hidden solid Removal Algorithm	Hidden solid Removal Algorithm	Learning about Hidden solid Removal Algorithm (B4, B5)
5.	Shading	Shading	Understanding Shading (B5, B6)
6.	Coloring	Coloring	Understanding Coloring (B5, B6)
7.	Computer Animation	Computer Animation	Learning about Computer Animation. (B4, B5)
8.	Computer Animation	Computer Animation	Learning about Computer Animation. (B4, B5)

VISUAL REALISM

Introduction

Visual Realism is a method for interpreting picture data fed into a computer and for creating pictures from difficult multidimensional data sets. Visualization can be classified as :

- Visualization in geometric modeling
- Visualization in scientific computing.

Visualization in geometric modeling is helpful in finding connection in the design applications. By shading the parts with various shadows, colors and transparency, the designer can recognize undesired unknown interferences. In the design of complex surfaces shading with different texture characteristics can use to find any undesired quick modifications in surface changes.

Visualization in computing is viewed as a technique of geometric modeling. It changes the data in numerical form into picture display, allowing users to view their simulations and computations. Visualization offers a process of seeing the hidden. Visualization in scientific computing is of great interest to engineers during the design process.

Existing visualization methods are:

- Parallel projections
- Perspective projection.
- Hidden line removal
- Hidden surface removal
- Hidden solid removal
- Shaded models

Hidden line and surface removal methods remove the uncertainty of the displays of 3D models and is accepted the first step towards visual realism. Shaded images can only be created for surface and solid models. In multiple step shading process, the first step is removing the hidden surfaces / solids and second step is shades the visible area only. Shaded images provide the maximum level of visualization.

The processes of hidden removal need huge amounts of computing times and also upper end hardware services. The creation and maintenance of such a models are become complex. Hence, creating real time images needs higher end computers with the shading algorithms embedded into the hardware.

Hidden line removal

Hidden line removal (HLR) is the method of computing which edges are not hidden by the faces of parts for a specified view and the display of parts in the projection of a model into a 2D plane. Hidden line removal is utilized by a CAD to display the visual lines. It is considered that information openly exists to define a 2D wireframe model as well as the 3D topological information. Typically, the best algorithm is required for viewing this information from an available part representation.

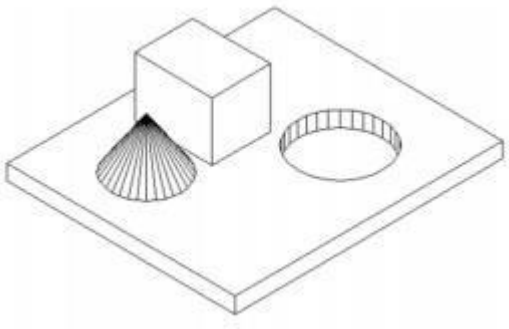


Fig 5.1: Hidden Line Removal

3D parts are simply manufactured and frequently happen in a CAD design of such a part. In addition, the degrees of freedom are adequate to show the majority of models and are not overwhelming in the number of constraints to be forced. Also, almost all the surface-surface intersections and shadow computations can be calculated analytically which results in significant savings in the number of computations over numerical methods.

1. Priority algorithm

Priority algorithm is basis on organization all the polygons in the view according to the biggest Z-coordinate value of each. If a face intersects more than one face, other visibility tests besides the Z-depth required to solve any issue. This step comprises purposes of wrapper.

Imagines that objects are modeled with lines and lines are generated where surfaces join. If only the visible surfaces are created then the invisible lines are automatically removed.

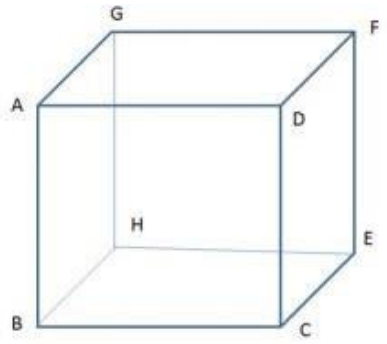


Fig 5.2: Priority Algorithm

Face	Priority
ABCE	1
ADFG	1
DCEF	1
ABHG	2
EFGH	2
BCEH	2

ABCD, ADFG, DCEF are given higher priority-1. Hence, all lines in this faces are visible, that is, AB, BC, CD, DA, AD, DF, FG, AG, DC, CE, EF and DF are visible.

AGHB, EFGH, BCEH are given lower priority-2. Hence, all lines in this faces other than priority-1 are invisible, that is BH, EH and GH. These lines must be eliminated.

Hidden surface removal

The hidden surface removal is the procedure used to find which surfaces are not visible from a certain view. A hidden surface removal algorithm is a solution to the visibility issue, which was one of the first key issues in the field of three dimensional graphics.

The procedure of hidden surface identification is called as hiding, and such an algorithm is called a 'hider'. Hidden surface identification is essential to render a 3D image properly, so that one cannot see through walls in virtual reality.

Hidden surface identification is a method by which surfaces which should not be visible to the user are prohibited from being rendered. In spite of benefits in hardware potential there is still a requirement for difficult rendering algorithms.

The accountability of a rendering engine is to permit for bigger world spaces and as the world's size approaches infinity the rendering engine should not slow down but maintain at constant speed.

There are many methods for hidden surface identification. They are basically a work out in sorting, and generally vary in the order in which the sort is executed and how the problem is subdivided. Sorting more values of graphics primitives is generally done by divide.

1. **Z - buffer algorithm**

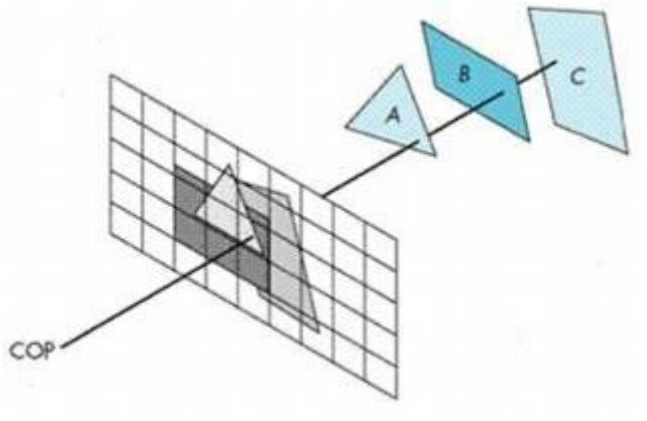


Fig 5.3: Z – buffer Algorithm

In Z-buffering, the depth of 'Z'value is verified against available depth value. If the present pixel is behind the pixel in the Z-buffer, the pixel is eliminated, or else it is shaded and its depth value changes the one in the Z-buffer. Z-buffering helps dynamic visuals easily, and is presently introduced effectively in graphics hardware.

- Depth buffering is one of the easiest hidden surface algorithms
- It keeps follow of the space to nearest object at every pixel position.
- Initialized to most negative z value.
- when image being drawn, if its z coordinate at a position is higher than z buffer value, it is drawn, and new z coordinate value is stored; or else, it is not drawn
- If a line in three dimensional is being drawn, then the middle z values are interpolated: linear interpolation for polygons, and can calculate z for more difficult surfaces.

Algorithm: *loop on y;*

loop on x;

zbuf[x,y] = infinity;

loop on objects

{

loop on y within y range of this object

{

loop on x within x range of this scan line of this object

{

if z(x,y) < zbuf[x,y] compute z of this object at this pixel & test zbuf[x,y] = z(x,y) update z-buffer

image[x,y] = shade(x,y) update image (typically RGB)

}

}

}

Basic operations:

1. compute y range of an object
2. compute x range of a given scan line of an object
3. Calculate intersection point of a object with ray through pixel position (x,y).

2. Painter'salgorithm

The painter's algorithm is called as a priority fill, is one of the easiest results to the visibility issue in three dimensional graphics. When projecting a 3D view onto a 2D screen, it is essential at various points to be finalized which polygons are visible, and which polygons are hidden.

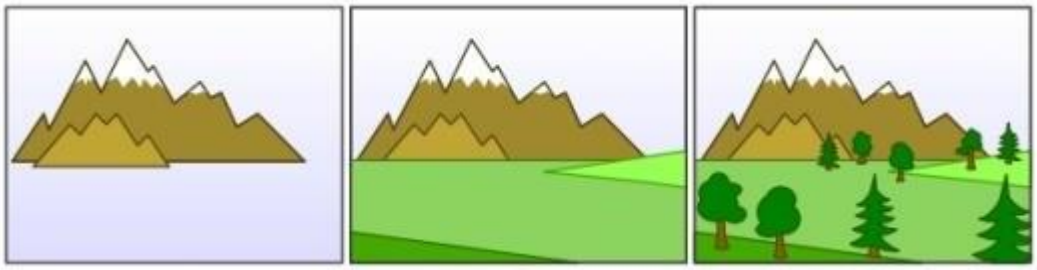


Fig 5.4: Painter's Algorithm

The 'painter's algorithm' shows to the method employed by most of the painters of painting remote parts of a scene before parts which are close thereby hiding some areas of distant parts. The painter's algorithm arranges all the polygons in a view by their depth and then paints them in this order, extreme to closest. It will paint over the existing parts that are usually not visible hence solving the visibility issue at the cost of having painted invisible areas

of distant objects. The ordering used by the algorithm is referred a 'depth order', and does not have to respect the distances to the parts of the scene: the important characteristics of this ordering is, somewhat, that if one object has ambiguous part of another then the first object is painted after the object that it is ambiguous. Thus, a suitable ordering can be explained as a topological ordering of a directed acyclic graph showing between objects.

Algorithm:

sort objects by depth, splitting if necessary to handle intersections; loop on objects (drawing from back to front)

```
{
loop on y within y range of this object
    {
    loop on x within x range of this scan line of this object
        {
        image[x,y] = shade(x,y);
        }
    }
}
```

Basic operations:

1. compute ‘y’range of an object
2. compute ‘x’range of a given scan line of an object
3. compute intersection point of a given object with ray via pixel point (x,y).
4. evaluate depth of two objects, determine if A is in front of B, or B is in front of A, if they don’t overlap in xy, or if they intersect
5. divide one object by another object

Advantage of painter's algorithm is the inner loops are quite easy and limitation is sorting operation.

3. Warnock algorithm

The Warnock algorithm is a hidden surface algorithm developed by John Warnock that is classically used in the area of graphics. It explains the issues of rendering a difficult image by recursive subdivision of a view until regions are attained that is trivial to evaluate. Similarly, if the view is simple to compute effectively then it is rendered; else it is split into tiny parts which are likewise evaluated for simplicity. This is a algorithm with run-time of O(np), where *p* is the number of pixels in the viewport and *n* is the number of polygons.

The inputs for Warnock algorithm are detail of polygons and a viewport. The good case is that if the detail of polygons is very simple then creates the polygons in the viewport. The continuous step is to divide the viewport into four equally sized quadrants and to recursively identify the algorithm for each quadrant, with a polygon list changed such that it contains polygons that are detectable in that quadrant.

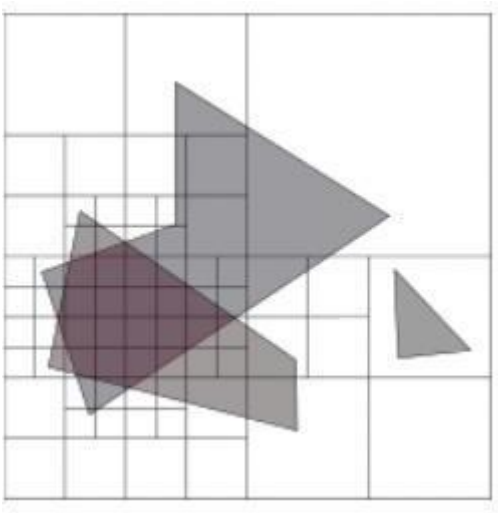


Fig.5.5. Warnock algorithm

1. Initialize the region.

2. Generate list of polygons by sorting them with their z values.

3. Remove polygons which are outside the area.

4. Identify relationship of each polygon.

5. Execute visibility decision analysis:

   a) Fill area with background color if all polygons are disjoint,

b) Fill entire area with background color and fill part of polygon contained in area with color of polygon if there is only one contained polygon,

c) If there is a single surrounding polygon but not contained then fill area with color of surrounding polygon.

d) Set pixel to the color of polygon which is closer to view if region of the pixel (x,y) and if neither of (a) to (d) applies calculate z- coordinate at pixel (x,y) of polygons.

6. If none of above is correct then subdivide the area and Go to Step 2.

Hidden Solid Removal

The hidden solid removal issue involves the view of solid models with hidden line or surface eliminated. Available hidden line algorithm and hidden surface algorithms are useable to hidden solid elimination of B-rep models.

The following techniques to display CSG models:

1. Transfer the CSG model into a boundary model.

2. Use a spatial subdivision strategy.

3. Based on ray sorting.

1. Ray-Tracing algorithm

A ray tracing is a method for creating an image by tracing the path of light via pixels in an image plane and reproducing the effects of its meets with virtual objects. The procedure is capable of creating a high degree of visual realism, generally higher than that of usual scan line techniques, but at a better computational. This creates ray tracing excellent suited for uses where the image can be rendered gradually ahead of time, similar to still images and film and TV visual effects, and more badly suited for real time environment like video games where speed is very important. Ray tracing is simulating a wide range of optical effects, such as scattering, reflection and refraction.

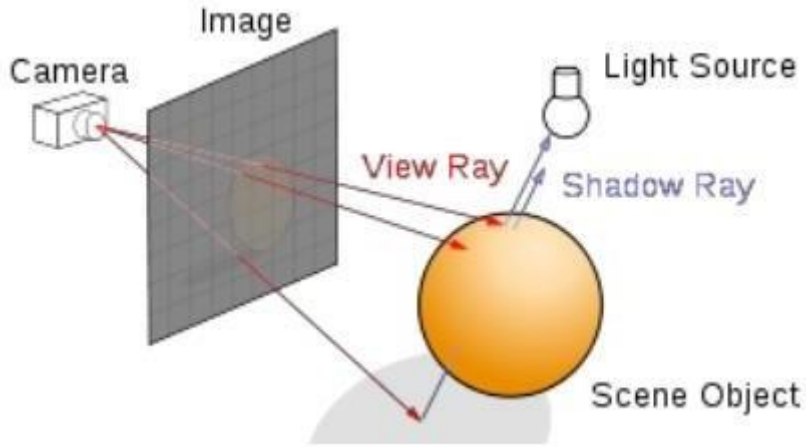


Fig.5.6. Ray-Tracing algorithm

Ray-Tracing algorithm

For every pixel in image

{

Generate ray from eye point passing via this pixel Initialize Nearest 'T'to 'INFINITY'

Initialize Nearest Object to NULL

For each object in scene

{

If ray intersects this image

{

If t of intersection is less than Nearest T

{

Set Nearest T to t of the intersection

Set Nearest image to this object

}

}

}

If Nearest image is NULL

{

Paint this pixel with background color

}

Else

{

Shoot a ray to every light source to check if in shadow

If surface is reflective, generate reflection ray

If transparent, generate refraction ray

Apply Nearest Object and Nearest T to execute shading function

Paint this pixel with color result of shading function

}

}

Optical ray tracing explains a technique for creating visual images constructed in three dimensional graphics environments, with higher photorealism than either ray casting rendering practices. It executes by tracing a path from an imaginary eye via every pixel in a virtual display, and computing the color of the object visible via it.

Displays in ray tracing are explained mathematically by a programmer. Displays may also incorporate data from 3D models and images captured like a digital photography.

In general, every ray must be tested for intersection with a few subsets of all the objects in the view. Once the nearest object has been selected, the algorithm will calculate the receiving light at the point of intersection, study the material properties of the object, and join this information to compute the finishing color of the pixel. One of the major limitations of algorithm, the reflective or translucent materials may need additional rays to be re-cast into the scene.

Advantages of Ray tracing:

1. A realistic simulation of lighting over other rendering.

2. An effect such as reflections and shadows is easy and effective.

3. Simple to implement yet yielding impressive visual results.

Limitation of ray tracing:

Scan line algorithms use data consistency to divide computations between pixels, while ray tracing normally begins the process a new, treating every eye ray separately.

Shading

Shading defines to describe depth perception in three dimensioning models by different levels of darkness. Shading is applied in drawing for describes levels of darkness on paper by adding media heavy densely shade for darker regions, and less densely for lighter regions.

There are different techniques of shading with cross hatching where perpendicular lines of changing closeness are drawn in a grid pattern to shade an object. The closer the lines are combining, the darker the area appears. Similarly, the farther apart the lines are, the lighter the area shows.

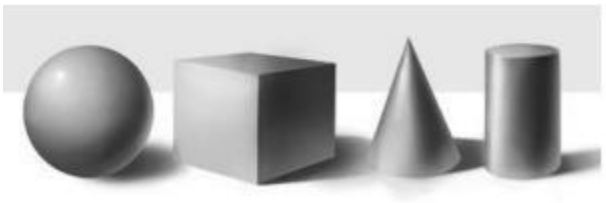


Fig.5.7. Shading

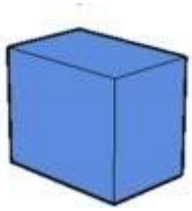


Fig.5.8. Image with edge lines

The image shown in figure 5.8 has the faces of the box rendered, but all in the similar color. Edge lines have been rendered here as well which creates the image easier to view.

Fig.5.9. Image without edge lines

The image shown in figure 5.9 is the same model rendered without edge lines. It is complicated to advise where one face of the box ends and the next starts.

Fig.5.10. Image with Shading

The image shown in figure 5.10 has shading enabled which makes the image extra realistic and makes it easier to view which face is which.

Shading techniques:

In computer graphics, shading submits to the procedure of changing the color of an object in the 3D view, a photorealistic effect to be based on its angle to lights and its distance from lights. Shading is performed through the rendering procedure by a program called a 'Shader'. Flat shading and Smooth shading are the two major techniques using in Computer graphics.

Goal: To use the lighting and reflectance model to shade facets of a polygonal mesh — that is, to assign intensities to pixels to give the impression of opaque surfaces rather than wireframes.

Assume we're given the following:

- $\bar{e}^w$ - center of projection in world coordinates
- $\bar{l}^w$ - point light source location
- I_a, I_d - intensities of ambient and directional light sources
- r_a, r_d, r_s - coefficients for ambient, diffuse, and specular reflections
- α - exponent to control width of highlights

Flat Shading

With flat shading, each triangle of a mesh is filled with a single color.

For a triangle with counterclockwise vertices $\bar{p}_1$, $\bar{p}_2$, and $\bar{p}_3$, as seen from the outside, let the midpoint be

$$\bar{p} = \frac{1}{3}(\bar{p}_1 + \bar{p}_2 + \bar{p}_3)$$

with normal

$$\vec{n} = \frac{(\bar{p}_2 - \bar{p}_1) \times (\bar{p}_3 - \bar{p}_1)}{\|(\bar{p}_2 - \bar{p}_1) \times (\bar{p}_3 - \bar{p}_1)\|}.$$

Then we may find the intensity at $\bar{p}$ using the Phong model and fill the polygon with that:

$$E = \tilde{I}_a r_a + r_d \tilde{I}_d \max(0, \vec{n} \cdot \vec{s}) + r_s \tilde{I}_d \max(0, \vec{r} \cdot \vec{c})^{\alpha},$$

where $\vec{s} = \frac{\bar{l}^w - \bar{p}}{\|\bar{l}^w - \bar{p}\|}$, $\vec{c} = \frac{\bar{e}^w - \bar{p}}{\|\bar{e}^w - \bar{p}\|}$, and $\vec{r} = -\vec{s} + 2(\vec{s} \cdot \vec{n})\vec{n}$.

Flat shading is a simple approach to filling polygons with color, but can be inaccurate for smooth surfaces, and shiny surfaces. For smooth surfaces—which are often tesselated and represented as polyhedra, using flat shading can lead to a very strong faceting effect. In other words, the surface looks very much like a polyhedron, rather than the smooth surface it's supposed to be. This is because our visual system is very sensitive to variations in shading, and so using flat shading makes faces really look flat.

Interpolative Shading

The idea of interpolative shading is to avoid computing the full lighting equation at each pixel by interpolating quantites at the vertices of the faces.

Given vertices $\bar{p}_1$, $\bar{p}_2$, and $\bar{p}_3$, we need to compute the normals for each vertex, compute the radiances for each vertex, project onto the window in device coordinates, and fill the polygon using scan conversion.

There are two methods used for interpolative shading:

Gouraud Shading The radiance values are computed at the vertices and then linearly interpolated within each triangle. This is the form of shading implemented in OpenGL.

Phong shading The normal values at each vertex are linearly interpolated within each triangle, and the radiance is computed at each pixel. Gouraud shading is more efficient, but Phong shading is more accurate. When will Gouraud shading give worse results?

Shading in OpenGL

OpenGL only directly supports Gouraud shading or flat shading. Gouraud is enabled by default, computing vertex colors, and interpolating colors across triangle faces. Flat shading can be enabled with

glShadeModel(GL FLAT).

This renders an entire face with the color of a single vertex, giving a faceted appearance.

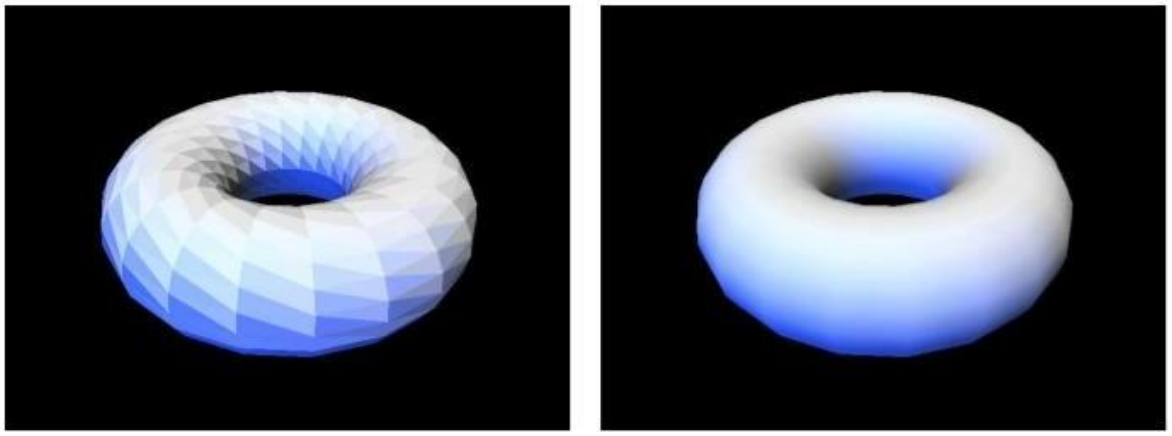


Fig 5.11: Left: Flat shading of a triangle mesh in OpenGL. Right: Gouraud shading.

Note that the mesh appears smooth, although the coarseness of the geometry is visible at the silhouettes of the mesh.

With pixel shaders on programmable graphics hardware, it is possible to achieve Phong shading by using a small program to compute the illumination at each pixel with interpolated normals. It is even possible to use a normal map to assign arbitrary normals within faces, with a pixel shader using these normals to compute the illumination.

COLORING

Color plays an important role in images. The light enters our eyes in the form of a spectrum with variation in the wavelengths. Human beings have three kinds of cones in their eyes that respond to different colors with different wavelengths. The basic colors red, green and blue represents long, medium and short wavelengths respectively. Introduction

The human vision is three-dimensional known as principle of trichromacy. According to this, the spectrum of light is encoded into three values that correspond to the amount of light absorbed by each type of cone.

Color Models

Color is a complex, interdisciplinary phenomenon spanning from psychology to physics. As the name suggests, color model is a model of colors. It can be defined as an organized system, which is used for the creation of a wide variety of colors starting from a small collection of primary colors.

Color models are of two types, as depicted in the figure 5.12.

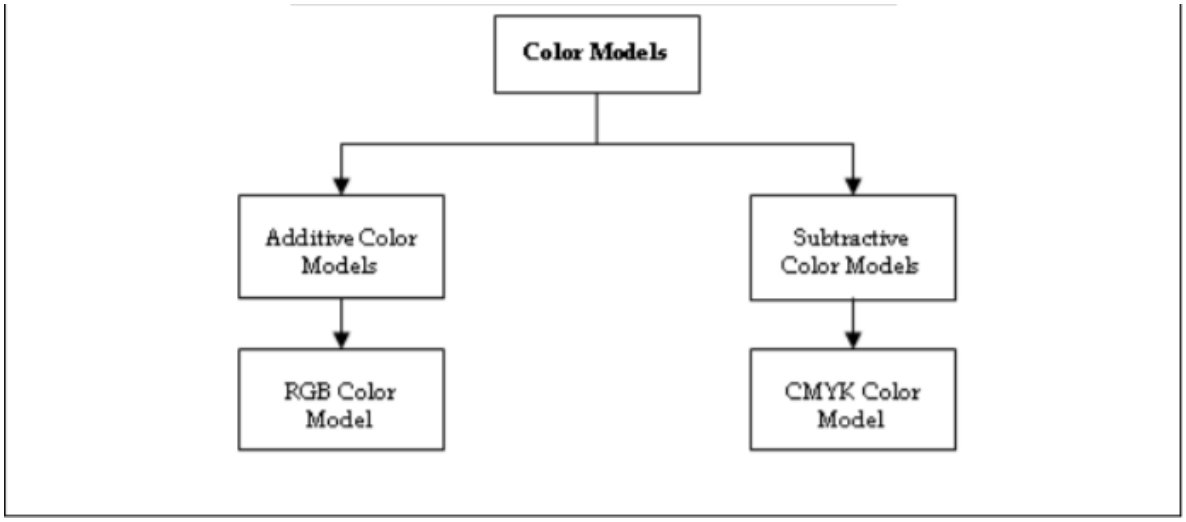


Fig 5.12: Color Models

1. **Additive Color Models:**
   Additive color models are those color models, which use light for displaying colors. In these color models, colors are seen as an aftereffect of transmitted light.

   The RGB color model is an example of an additive color model.
2. **Subtractive Color Models:**
   Subtractive color models are those color models, which use printing inks for displaying colors. In these color models, colors are seen as an aftereffect of reflected light.
   The CMYK color model is an example of a subtractive color model.

RGB Color Model

The RGB color model is an additive color model, which uses three primary colors, namely, red (R), green (G), and blue (B). These primary colors can further be used to create secondary colors, such as yellow, white, and magenta. Computers can display approximately 16.7 million colors using this color model.

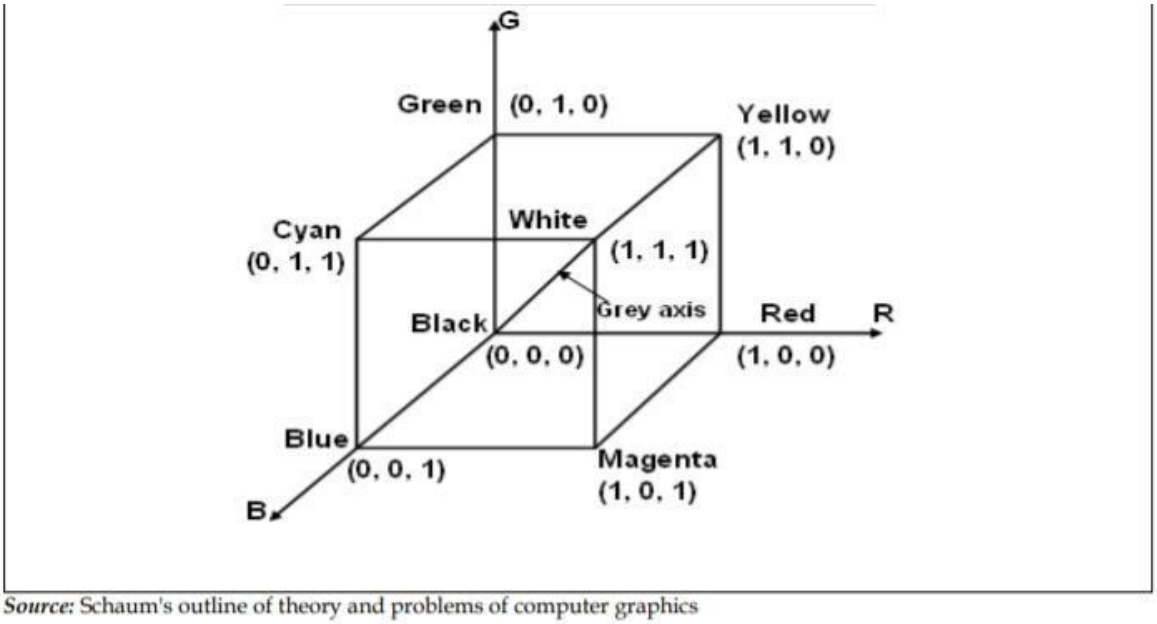
In RGB model, every color's measurement is done with a value that ranges from 0 to 255, where 0 means no light, and value 255 means maximum intensity. This (0 to 255) is the limit of storing information in 1 byte of computer memory. In RGB model,

3 bytes (24 bits) of information are needed for defining the three primary colors RGB.

A large percentage of the visible band of colors can be represented by mixing red, green, and blue (RGB) colored light in various intensities and proportions. RGB colors are also called as additive colors, as they combine to create white color.

In other words, adding all colors creates white, that is, all visible wavelengths are transmitted back to the eye. Additive colors are used for monitors, video, and lighting.

A unit cube is used to represent the color space of RGB colors. This unit cube is placed at origin in its standard position. The figure 5.13 depicts the color co-ordinate system with the three primary colors of RGB model, that is, RGB.



Source: Schaum's outline of theory and problems of computer graphics

Fig 5.13: Color Coordinate System with RGB Colors

Each of the RGB (primary) colors can take up the intensity value ranging from 0 being the lowest to 1 being the highest. When these primary colors are mixed at different intensity levels, they produce a variety of colors including cyan, magenta, yellow, and white. The collection of all the colors available by such a linear combination of red, green, and blue forms the cube-shaped RGB color space as shown in figure 5.13.

The corner of the RGB color cube that is at the origin of the co-ordinate system corresponds to black. The corner of the cube that is diagonally or obliquely opposite to the origin represents white. The oblique line connecting white and black corresponds to all the gray colors between white and black and it is called as the gray axis. With the help of this RGB color model, a random color within the cubic color space can be specified by its color co- ordinates, that is red, green, and blue.

(0,0,0) is for black, (1,1,1) is for white, (1,1,0) for yellow.

The color specification using the RGB color model is an additive process. First, black color is considered and the suitable primary component is added on to yield a preferred color.

CMYK Color Model

The colors cyan (C), magenta (M), yellow (Y) and black (K) (CMYK) is a subtractive color model, which is mainly used in color print production. This color model is also called as acomplementary RGB model.

In this color model, inks of four colors, cyan (C), magenta (M), yellow (Y) and black (K) are used for defining colors. Every color has some ink amount, whose measurement is done by using percent starting from 0 to 100, where a value of 100 signifies the application of inks at full intensity. This CMY color model describes colors using a subtractive process, which closely matches the working principles of the printer.

The color space of CMYK color model is called subtractive. The application of inks of cyan, magenta, yellow, and black color to a surface that is white in color, is done for subtracting some color from that white surface. This method of subtraction finally leads to the creationof the final color. This model is generally called as CMY model.

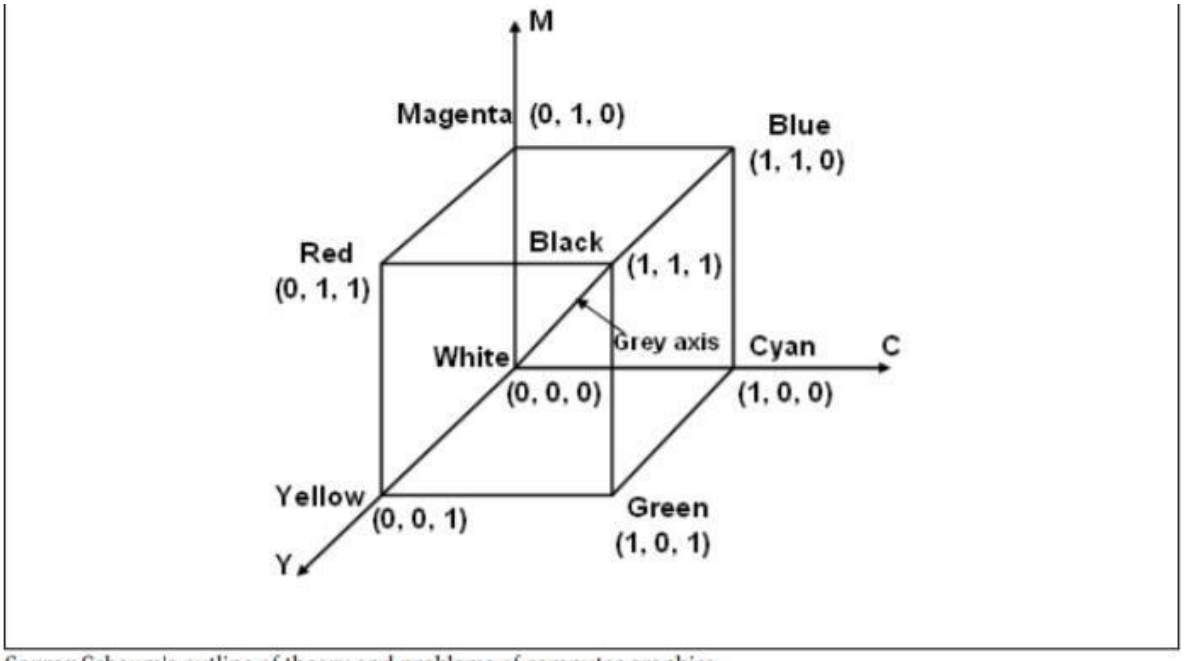
The subtraction of all colors by the CMY combination at full intensity renders black color. Nevertheless, because of impurities that exist in the available CMY inks, complete and equal intensity is made impossible, and filtering of some RGB light takes place, which renders a muddy brown color. Thus, with the addition of the black ink CMY color model takes place.

In the CMY model, first the white color is considered, and the suitable primary components are taken away, to yield a preferred color.

Example: If we subtract or decrease the red color from white color, the color that remains consists of green and blue, which is nothing but the cyan color.

If the above example is considered from another perspective, the quantity of cyan, the complementary color of red, can be used in order to control the amount of red, which isequal to one minus the amount of cyan color.

The following figure 5.14 depicts a co-ordinate system using the three primary complementary colors which are cyan (C), magenta (M), and yellow (Y).



Source: Schaum's outline of theory and problems of computer graphics

Fig 5.14: Color Co-ordinate System with CMY Colors

In the figure 5.14, the corner of the CMY color cube, that is, at (0, 0, 0) represents white color, whereas the corner of the cube that is at (1, 1, 1) represents black color. CMY color model is mainly used for the production of printed materials. This is the reason why CMY color model is an important element in printing.

COMPUTER ANIMATION

Introduction

Visual effect is part of advanced computer graphics that pertains to the various processes through which images are created and manipulated in the absence of a live environment. When the sets are costly to setup and when it becomes impossible to capture a film or object, visual effects come into picture. They provide realistic image of the live environment. The images in visual effects are computer generated and can be seen in most of the movies. Visual effects provide more appeal to the story line of a movie and are generally completed after the completion of movie but are pre-planned before the commencement of a movie shoot. Visual effects employ graphic designing, modeling, animation and other relevant software but special effects are made on the sets. Today, one can find the application of computer graphics and visual effects and animation in diverse fields such as science, technology, military, advertising, entertainment, and so on.

Animation

The word animation is derived from the Latin word ‘anima’ which is the animating principle. It is used as a translation for the Greek word psyche and relates to the Christian concept of soul. Hence, animation can be defined as a technique of giving soul to drawings and art work which are lifeless. Earliest examples of animation can be traced back to the Paleolithic cave paintings where attempts were made to capture the phenomenon of motion drawing.

The term animation is used to display a sequence of images rapidly employing either 2-D or 3-D art works or positions of various models to bring out an illusionary movement to the models and art work. This is done through optical illusion of motion by employing the phenomenon of persistence of vision in which images can be created and depicted in several ways. Motion picture or video programs are the commonly used methods of animation.

Persistence of Vision

Persistence of vision is a phenomenon of the eye where an afterimage is thought to persist for one twenty-fifth of a second on the retina of the eye. In the visual perception phenomenon, the eye is not considered as a camera. The phi phenomenon has a more constructionist approach towards cinema while persistence of vision is a realistic approach.

Today, persistence of vision is an accepted phenomenon in the history of cinema. In the early days, it was determined that a frame rate of less than 16 frames per second had enabled the mind to see flashing images. Audiences, sometimes interpret motion at ten frames per second or even slower. The modern theatrical films run at 24 frames per second.

The two perceptual illusions are:

1. Phi Phenomenon: This method is an optical illusion defined by Max Wertheimer in the Gestalt psychology in 1912. It is based on the principle that the human eye is able to perceive movements based on pieces of information. This is similar to the succession of images. The speed of the images per second can be observed for the phi phenomenon. The figure 5.15 shows the phi phenomenon and depicts how an object moves in a circular motion with constant speed.

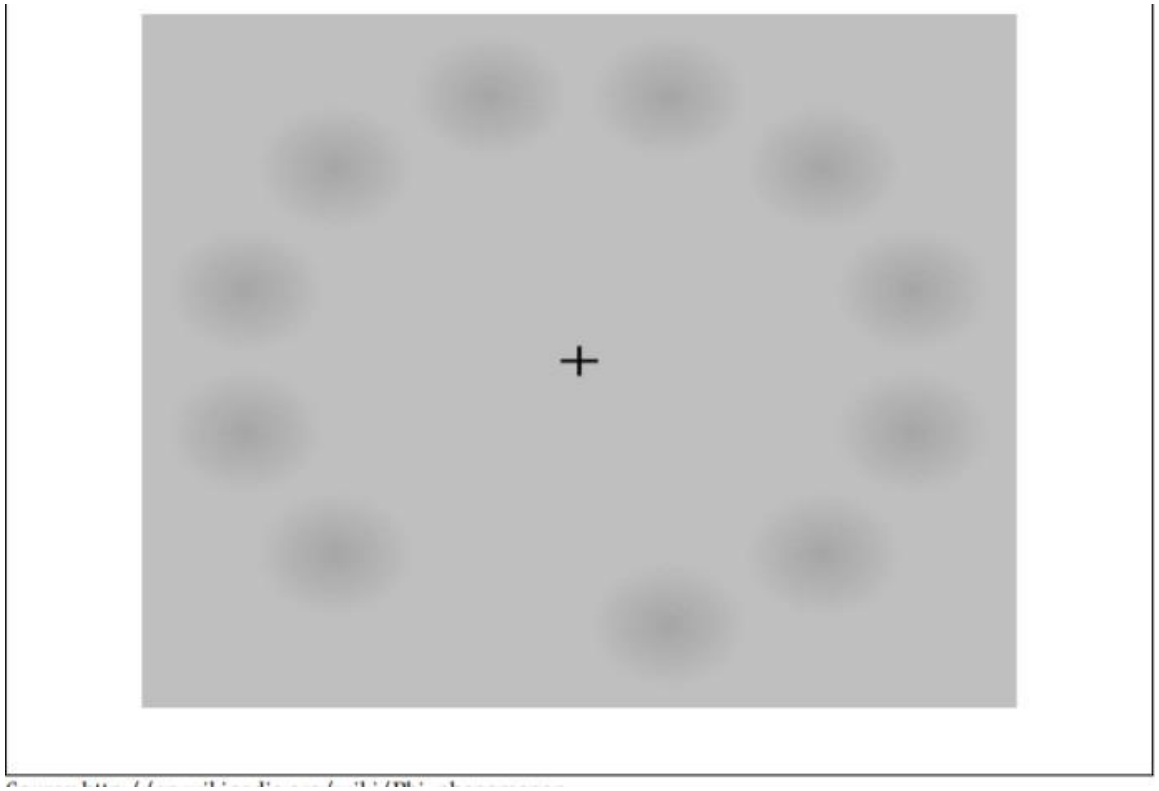


Fig 5.15: Phi Phenomenon

2. Beta Movement: The beta movement is considered to be a perceptual illusion where the brain merges two more still images, imaginatively to be in motion. This phenomenon creates an illusion of motion both towards and away from the viewer. Consider phenomenon of a viewer watching a screen. This screen is projected with two images in succession. The first image shows a ball on the left side of the frame, while the second image shows a ball on the right side of the frame. These images are shown in rapid succession. Thus, as a viewer you can see one ball moving from the left towards the right but not two balls flashing in succession. When the two objects are of different size that is when the first image is large and the second image is small. As a viewer, you can only view that the object is moved away from us.

The following figure 5.16 depicts the beta phenomenon and provides a clear understanding about how the image keeps changing its position.

Fig 5.16: Beta Phenomenon

Animation came into existence with the advent of cinematography. Georges Melies is considered as the creator of special effect films and is also the first person to have used animation with special effects. George Melies came up with the stop motion animation. In this method, the rolling camera was stopped to make changes in the scene and then continued to roll the film.

This technique was discovered accidently when George Melies' camera broke while shooting a moving bus. Immediately after this incident, George restarted shooting; this time he happened to shoot a hearse pass by. Finally, George observed that he had transformed
the moving bus into a moving hearse. This made George analyze about what he had created. This concept was later used in the field of animation and George Melies is considered to be a great contributor to the field of animation.

Animated films are also referred to as cartoons, which is an industry of its own today. John Randolph Bray is a successful producer of animations along with Earl Hurd who came up with the idea of cel animation process.

The photographs of the drawings are drawn on paper. To create an illusion of movement the drawings are done in such a way that they slightly differ from the previous one. Then these drawings are photocopied onto transparent acetate sheets called cels. These cels are filled with paints and are assigned colors on the opposite side of the line drawings. The cels are then photographed one by one onto motion picture film against a painted background by a rostrum camera. Cel animation is obsolete now.

Today, drawings with their backgrounds are scanned using a computer system. Software programs are used to color the drawings and their effects are simulated.

Nowadays, a complete animated film is done in a variety of styles such as the films produced by Walt Disney to the Cartoon Styles produced by the Warner Brothers.

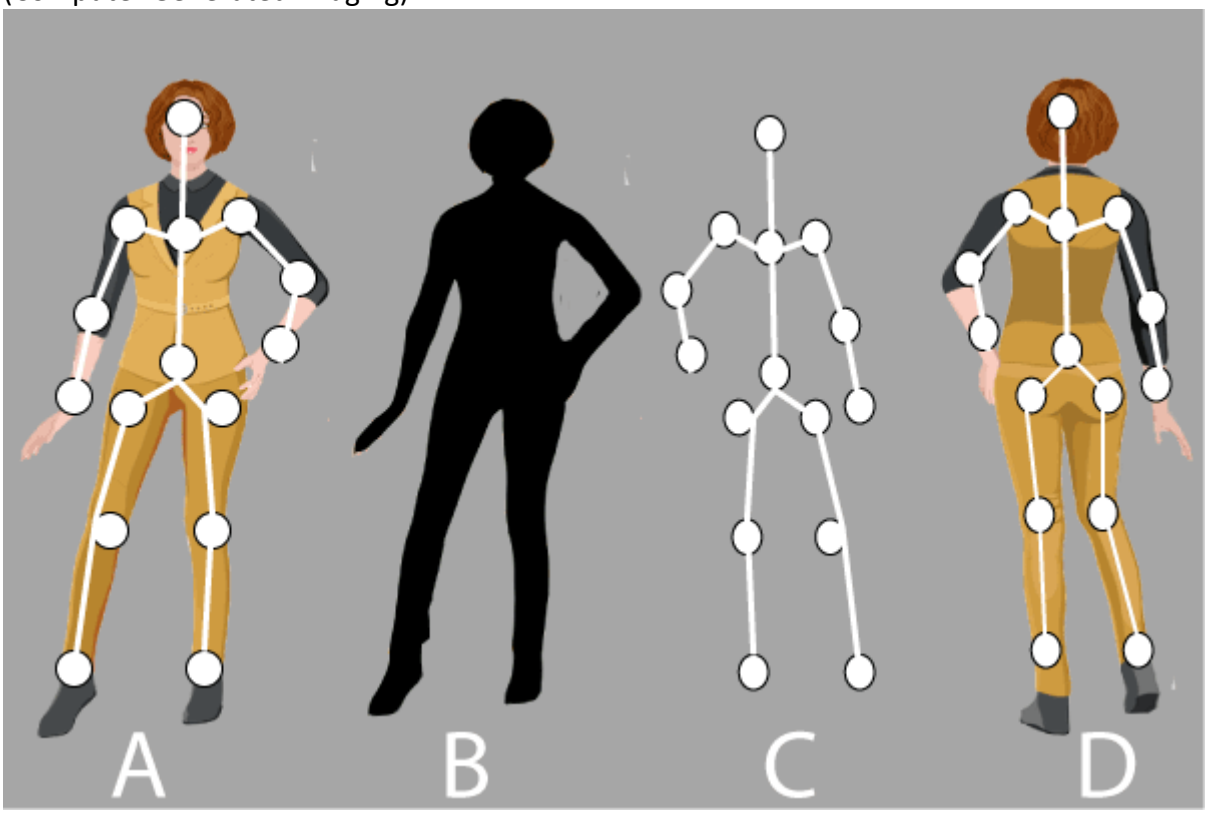
Computer Animation

Computer animation uses a variety of techniques to be created digitally on a computer. The following are some of the techniques:

1. 2-D Animation: In 2-D animation, figures are created and edited on the computer by either using 2-D bitmap graphics or by using 2-D vector graphics. This method uses the traditional animation techniques like tweening, morphing, onion skinning, and interpolated rotoscoping. Foster's Home for Imaginary Friends, Danny Phantom, Waltz with Bashir, The Grim Adventures of Billy and Mandy are some of the examples for 2-D films.

2. 3-D Animation: In 3-D animation the figures are digitally modeled and manipulated using an animator. A process known as rigging is carried out. In this process to manipulate a mesh, a digital skeletal structure is used to control the mesh. Techniques like mathematical functions, simulated fur or hair, effects such as fire and water and the use of motion capture, fall under the 3- D dynamics. Sometimes it is difficult to distinguish between 3-D animations and live actions. In such cases visual effects are used.

Computer Animation is an art to produce moving images. We can define it as a subfield of computer graphics and animation. When we use animation in a movie or film, it is called CGI (Computer Generated Imaging).



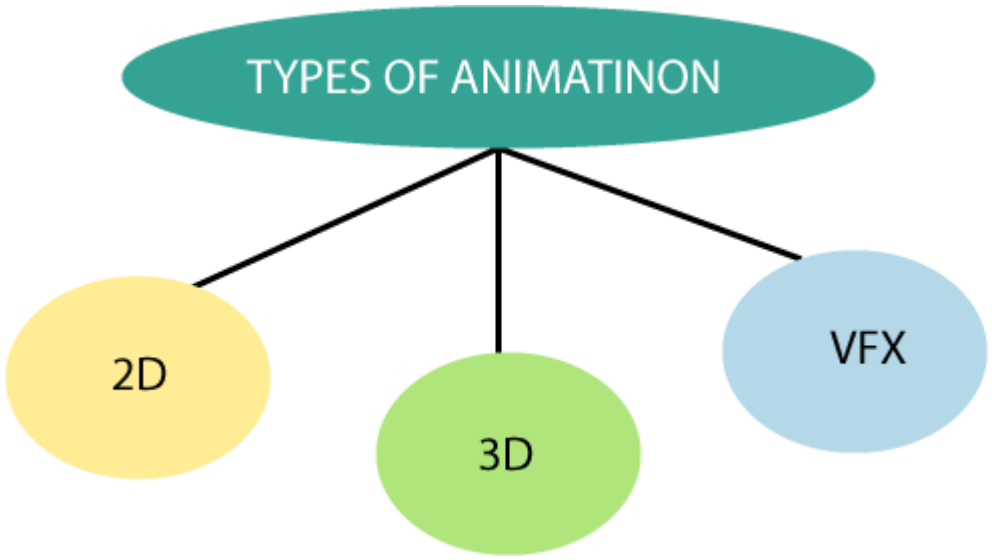
Application Areas of Animation

Animation is used in various fields. They are:

1. **Education:** An animated video or animated tutorial is used to increase the learning of the students. The animation is mostly used in schools, colleges, and training centers.
2. **Entertainment:** It is a vast area where we can use animations. Various animation methods are used in cartoons, movies, motion pictures, and TV shows.
3. **Advertisement:** Animation is also used to make ad films that take less space and captures the vast attention of the people. It is a better way to provide more information about products.
4. **Medical:** Animation is used in medical science. The professionals and the students can easily visualize the details of human anatomy through medical animation.
5. **Retail:** The marketers use animation to present the attributes of products to the customers.
6. **Computer-Aided Design:** Animation plays an important role in CAD. We can use CAD in automobile designing and many other designing.
7. **Gaming:** Animation is widely used in gaming. We can develop 3D games by using animation.
8. **Presentation:** We can present anything through an animated presentation. Animated presentation is a better way to represent an idea.

Types of Animation

The Animation is divided into three parts.



- **2D:** It is also called **"2 Dimensional."**2D animation is defined as a process of producing characters, storyboards, and backgrounds in a two-dimensional environment.

2D animation is a flat animation that only works on two coordinates x-axis(horizontal) and y-axis(vertical).

The 2D animation uses raster and vector graphics to produce and edit animated images.

For Example– Cartoons (Daffy Duck, Snow White, the jungle book, etc.)

The software used in 2D animation is:

1. Adobe Photoshop
2. Adobe Flash
3. After Effects
4. Encore

2D

Advantages:

1. Easy to Control.
2. Less Time-Consuming.
3. Less Production Cost.
4. Easy to change.

Disadvantages:

1. Need proper skills to use animation software.

- **3D:** It is also called **"3 Dimensional."** 3D animation is defined as the process of producing three-dimensional motion images in the digital environment. 3D Animation is used to create animated scenes through the computer.

We can manipulate 3D objects or models through the 3D animation software.
3D animation works on all three coordinates x-axis(height), y-axis(width) and z-axis(depth).

For Example–

1. Toy Story
2. Jurassic Park
3. The Transformers

2D

3D

The software used in 3D animation is:

1. ZBrush
2. Houdini
3. Blender
4. Autodesk Maya

The process of producing 3D animations is divided into three parts:

1. **Modeling:** It is a phase in which we can describe the creation of 3D objects within a specific scene.
2. **Layout and Animation:** It is a phase in which we can describe how objects are positioned and animated within a scene.
3. **Rendering:** It is a final process. It is a phase in which we can convert 3D objects or images into 2D objects or images.

Advantages:

1. Motion Communication
2. More Realistic
3. Reusable

Disadvantages:

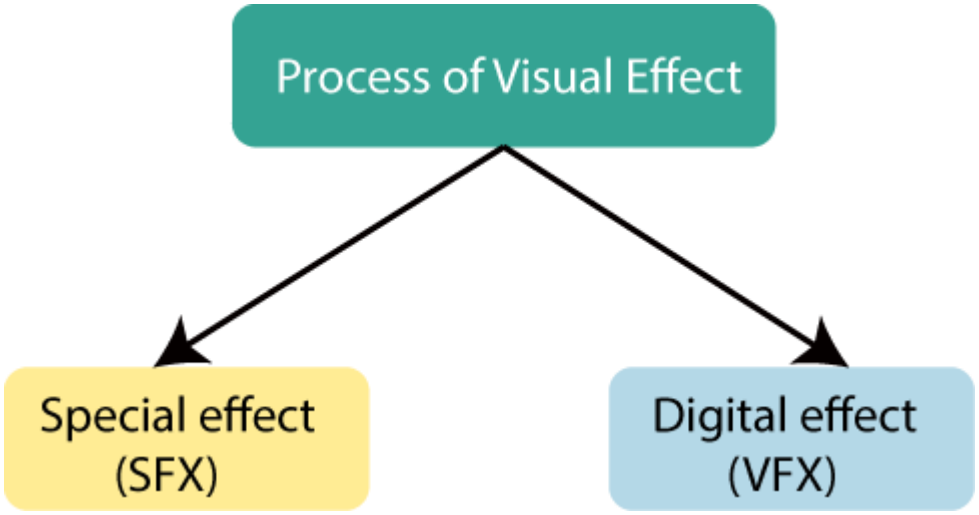
1. Lack of purity
2. Limited Thought and Ideas
3. **VFX:** It is also called **"Visual Effects."** It is a process in which we can make a mixture of real shooting (special effects) with animated images (digital effects). It is mostly and widely used in movie making.

For Example-
In a movie, we will see the hero flying into the air by using VFX effects.
In the Jurassic Park movie, the dinosaurs are also created by using VFX effects.

The Software used in VFX animation:

1. Nuke
2. After effects
3. 3D Studio Max
4. Adobe Creative Suite 4

This process has been done in two parts-



- **Special effects (SFX):** The special effects cover all visual effects that occur in live-action.

For Example- Stunt, Any Explosion on set.

- **Digital Effects (VFX):** It is also called **"Visual effects."** It includes different steps by which photographic assets can create, and imagery can manipulate.

Digital effects mostly include a combination of real photography and computer-aided imagery to create a realistic environment. The environment would be impossible, costly, and sometimes very dangerous to shoot in-camera.

Advantages:

1. It helps to show unreal objects.
2. It saves a lot of time.

Disadvantages:

1. Need a Lot of Planning.
2. Overuse of Visual Effects

Tutorial Questions

1. Explain about visual realism
2. Write Algorithm for hidden line removal.
3. Write Algorithm for hidden surface removal.
4. Write Algorithm for hidden solid removal.
5. Explain about shading.
6. Explain about coloring.
7. Explain about computer animation and its applications.

QUESTION BANK

1. Write Algorithm for hidden line removal.
2. Write Algorithm for hidden surface removal.
3. Write Algorithm for hidden solid removal.
4. Explain about shading and coloring.
5. Explain about computer animation and its applications